



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**NUMERICAL SIMULATION INVESTIGATIONS
IN WEAPON DELIVERY PROBABILITIES**

by

Kristofer A. Peterson

June 2008

Thesis Advisor:

Morris Driels

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2008	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE Numerical Simulation Investigations in Weapon Delivery Probabilities			5. FUNDING NUMBERS	
6. AUTHOR(S) Kristofer Peterson				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) <p>The study of weapon delivery probabilities has historically been focused around analytical solutions and approximations for weapon delivery accuracy and effectiveness calculations. With the relatively recent increase in modern computing power many of the historical expressions can be simulated quickly with similar or more accurate results than the historical expressions and approximations.</p> <p>In this thesis simulation methods are used to evaluate weapon delivery probability parameters including circular error probable, range and deflection error probable, and weapon effectiveness in the single and salvo weapon scenarios. Comparisons of the simulation results and corresponding historical practices are made to validate simulation techniques.</p> <p>Additionally, standard deviations in the range and deflection direction are extracted from weapon impact data. Using these extracted standard deviations weapon effectiveness calculations are performed.</p>				
14. SUBJECT TERMS Weapon Delivery Accuracy, Weapon Effectiveness, Salvo Formula, Aiming Error, Ballistic Dispersion, Error Probable, Probability Simulation			15. NUMBER OF PAGES 85	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

**NUMERICAL SIMULATION INVESTIGATIONS
IN WEAPON DELIVERY PROBABILITIES**

Kristofer A. Peterson
Civilian, United States Air Force
B.S. Aeronautical Engineering, Embry-Riddle Aeronautical University, 1999

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
June 2008**

Author: Kristofer A. Peterson

Approved by: Morris Driels
Thesis Advisor

Anthony Healey
Chairman, Department of Mechanical and
Aeronautical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The study of weapon delivery probabilities has historically been focused around analytical solutions and approximations for weapon delivery accuracy and effectiveness calculations. With the relatively recent increase in modern computing power many of the historical expressions can be simulated quickly with similar or more accurate results than the historical expressions and approximations.

In this thesis simulation methods are used to evaluate weapon delivery probability parameters including circular error probable, range and deflection error probable, and weapon effectiveness in the single and salvo weapon scenarios. Comparisons of the simulation results and corresponding historical practices are made to validate simulation techniques.

Additionally, standard deviations in the range and deflection direction are extracted from weapon impact data. Using these extracted standard deviations weapon effectiveness, calculations are performed.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION TO WEAPONNEERING CONCEPTS	1
A.	PROBABILITY DENSITY FUNCTION	1
1.	Univariate Normal Distribution	4
a.	<i>Univariate Normal PDF</i>	<i>4</i>
b.	<i>Univariate Normal Cumulative Density Function (CDF)</i>	<i>4</i>
2.	Bivariate Normal Distribution.....	7
a.	<i>Bivariate Normal PDF.....</i>	<i>7</i>
b.	<i>Bivariate Normal CDF.....</i>	<i>7</i>
3.	Circular Normal and Rayleigh Distributions.....	7
a.	<i>Circular Normal PDF.....</i>	<i>7</i>
b.	<i>Rayleigh PDF.....</i>	<i>8</i>
c.	<i>Rayleigh CDF.....</i>	<i>8</i>
B.	ERROR TYPES	8
1.	Ballistic Dispersion.....	8
2.	Aiming Error.....	8
C.	ACCURACY	9
1.	Circular Error Probable (CEP).....	9
2.	Range Error Probable (REP) and Deflection Error Probable (DEP).....	9
3.	Relationship of CEP to REP/DEP	11
D.	COMBINING ERROR TYPES.....	11
1.	Single Round Scenario.....	12
2.	Salvo Scenario	13
3.	Simulation Implementation.....	15
II.	TEST DATA CHARACTERISTICS	17
A.	UNIVARIATE NORMAL.....	17
B.	RAYLEIGH.....	18
C.	SALVO FORMULA	19
III.	MAINTAINING AIMING ERROR AND BALLISTIC DISPERSION AS SEPARATE PARAMETERS	21
A.	ACCURACY CALCULATIONS	21
1.	Single Round Scenario.....	22
2.	Salvo Scenario	23
B.	WEAPON EFFECTIVENESS CALCULATIONS	26
1.	Single Round Scenario.....	27
2.	Salvo Scenario	29
IV.	SALVO EFFECTIVENESS CALCULATIONS: SIMULATION VS. ANALYTICAL APPROXIMATIONS	31
A.	CIRCULAR TARGET	31
B.	SQUARE TARGET	39

V.	ATTEMPTS TO EXTRACT ERROR TYPES FROM IMPACT DATA	41
A.	ALGORITHM DESCRIPTION	41
B.	SINGLE ROUND SCENARIO	42
1.	Double Normal Approximation to Non-Normal Dataset	42
a.	<i>Double Normal Dataset Weapon Effectiveness Calculations</i>	46
2.	Double Rayleigh Approximation to Non-Normal Dataset	47
a.	<i>Double Rayleigh Dataset Weapon Effectiveness Calculations</i>	49
VI.	CONCLUSIONS AND RECOMMENDATIONS	51
APPENDIX.	MATLAB CODE	53
A.	CHAPTER II CODE	53
1.	Code for Table 2	53
2.	Code for Table 3	53
3.	Code for Table 4	53
B.	CHAPTER III CODE	54
1.	Code for Table 5-Table 9	54
2.	Code for Table 10-Table 11	56
C.	CHAPTER IV CODE	57
1.	Code for Table 12-Table 25	57
2.	Code for Table 26	59
D.	CHAPTER V CODE	60
1.	Double Normal Approximation	60
2.	Double Rayleigh Approximation	63
	LIST OF REFERENCES	67
	INITIAL DISTRIBUTION LIST	69

LIST OF FIGURES

Figure 1	Sample Impact Points for 250 Bombs	2
Figure 2	Impact Histogram.....	3
Figure 3	50,000 Samples Impact Histogram	6
Figure 4	Circular Error Probable.....	9
Figure 5	Range Error Probable and Deflection Error Probable	10
Figure 6	Single Round: Aiming Error and Ballistic Dispersion	12
Figure 7	Single Round Scenario: Four Occasions.....	13
Figure 8	Salvo Scenario: Four Occasions; Five Bombs/Salvo.....	14
Figure 9	Salvo Effectiveness Simulation Flowchart [From 3].....	16
Figure 10	Flowchart For Accuracy Simulations	22
Figure 11	Flow Chart for Weapon Effectiveness Simulations.....	27
Figure 12	Flow Chart for Extracting Weapon Effectiveness From Impact Data.....	42
Figure 13	Artificially Created Double Normal CDF ($w_{x1} = 0.3, \sigma_{x1} = 30, \sigma_{x2} = 5$).....	44
Figure 14	Comparison of CDFs: Double Normal Dataset vs. Curve Fitting	45
Figure 15	Artificially Created Double Rayleigh CDF ($w_{r1} = 0.3, \sigma_{r1} = 30, \sigma_{r2} = 5$).....	48
Figure 16	Comparison of CDFs: Double Rayleigh Dataset vs. Curve Fitting.....	49

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1	MATLAB Code For Integrating the Non-standard Normal PDF	5
Table 2	REP Convergence Data.....	17
Table 3	CEP Convergence Data.....	18
Table 4	Salvo Formula Convergence Data	19
Table 5	Separate vs. RSS Errors for Accuracy Calculations (1 bomb per salvo).....	23
Table 6	Separate vs. RSS Errors for Accuracy Calculations (5 bombs per salvo)	24
Table 7	Separate vs. RSS Errors for Accuracy Calculations (10 bombs per salvo)	24
Table 8	Separate vs. RSS Errors for Accuracy Calculations (50 bombs per salvo)	25
Table 9	Separate vs. RSS Errors for Accuracy Calculations (100 bombs per salvo) ...	25
Table 10	Separate vs. RSS Errors for Weapon Effectiveness (1 bomb per salvo)	28
Table 11	Separate vs. RSS Errors for Weapon Effectiveness (5,10,50,100 bombs per salvo).....	29
Table 12	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=5$; Lethal Radius 5).....	31
Table 13	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=5$; Lethal Radius 10).....	32
Table 14	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=5$; Lethal Radius 20).....	32
Table 15	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=5$; Lethal Radius 30).....	33
Table 16	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=5$; Lethal Radius 40).....	33
Table 17	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=5$; Lethal Radius 50).....	34
Table 18	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=5$; Lethal Radius 60).....	34
Table 19	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=25$; Lethal Radius 5).....	35
Table 20	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=25$; Lethal Radius 10).....	35
Table 21	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=25$; Lethal Radius 20).....	36
Table 22	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=25$; Lethal Radius 30).....	36
Table 23	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=25$; Lethal Radius 40).....	37
Table 24	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=25$; Lethal Radius 50).....	37
Table 25	Salvo Sim. vs. Approximation: ($\sigma_{aiming}=50$; $\sigma_{bd}=25$; Lethal Radius 60).....	38
Table 26	Circular vs. Rectangle Weapon Effectiveness Area Salvo Simulation.....	39

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my thesis advisor Professor Morris Driels. His passion for weaponeering was a great source of encouragement for my research. Our morning meetings were always extremely valuable and I have no doubt the lessons learned will be with me throughout my career.

I would also like to thank my supervisor at Edwards AFB Mr. Tony Rubino. His leadership and dedication to my personal and professional development have given me opportunities I never would have thought possible. Thanks Tony.

Finally, I must thank my beautiful fiancé Ms. Jessica Ulvin. Without her love, support, and encouragement this year would have been an even greater challenge. I am looking forward to our future and facing new exciting challenges together.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION TO WEAPONEEING CONCEPTS

In general terms, weaponeeing is the process of determining the quantity of a specific type of weapon required to achieve a specific level of target damage, considering target vulnerability, weapon effects, munition delivery errors, damage criteria, probability of kill, weapon reliability, etc. [1] This thesis will focus on munition delivery error statistics and probability of hit for various scenarios. These topics are inherently random in nature requiring a statistical approach for analysis. The weaponeeing concepts discussed require a general understanding of some basic statistical definitions and methods. This chapter provides the necessary statistical background to follow the analysis in the following chapters.

A. PROBABILITY DENSITY FUNCTION

The probability density function (PDF) describes the likelihood that a random event will result in a certain value contained within a defined population. Flipping a coin is a classic example. There is an equal probability that the coins will show heads or tails. Because the probability density function must account for all possible outcomes the total sum of all possibilities must be one. It is intuitively obvious that the probability of “heads” resulting from a coin toss is $1/2$. The same is true for “tails”. This yields a sum of all probabilities equal to unity as expected. This is an example of a discrete PDF. Discrete meaning the data set consists of fixed values with discontinuous jumps for the results. A Coin flip or roll of a die are clear examples of discrete random processes. The results from these events can only be:

- Coin Toss: heads or tails
- Die roll: 1, 2, 3, 4, 5, or 6

Of more relevance to weaponeeing considerations is the continuous PDF. A probability can be obtained for any result within the bounds of the population. Take for example an aircraft dropping unguided bombs on a target. Suppose the aircraft drops 250 bombs with impacts as shown in Figure 1.

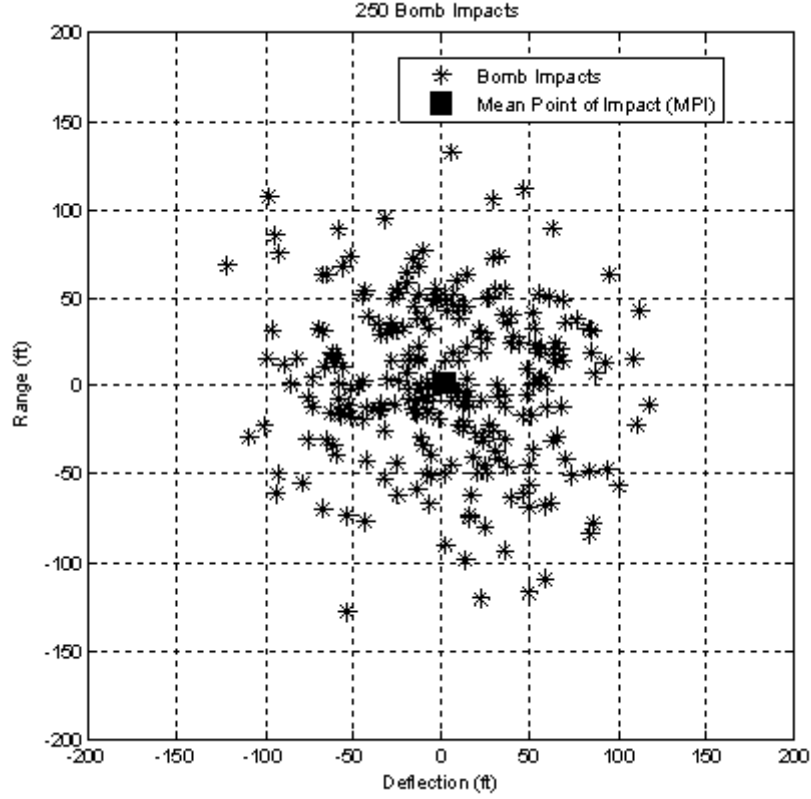


Figure 1 Sample Impact Points for 250 Bombs

Range is defined as the direction the aircraft is heading when the bomb is released. Deflection is perpendicular to the range direction with the origin of the system defined as the desired mean point of impact (DMPI). The impacts clearly display the randomness of this delivery. Two main parameters used to describe the dataset are the mean and variance. The mean, or \bar{x} , is the average of the parameters in the dataset and provides a relative location of the dataset to the target. Here the mean values for range and deflection are displayed by the mean point of impact (MPI). The variance is defined as:

$$S_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (1)$$

The standard deviation is defined as the square root of the variance and is labeled σ . Standard deviation represents a key parameter for characterizing weapon delivery accuracy and effectiveness and will be discussed extensively.

The randomness of the deliveries can be evaluated further by splitting each axis into bins and counting how many bombs are contained in each bin. For example, for the impact data in Figure 1.1 the deflection axis can be split into 15 ft. bins. Each bin is then evaluated to determine how many impacts it contains. Figure 2 displays these results.

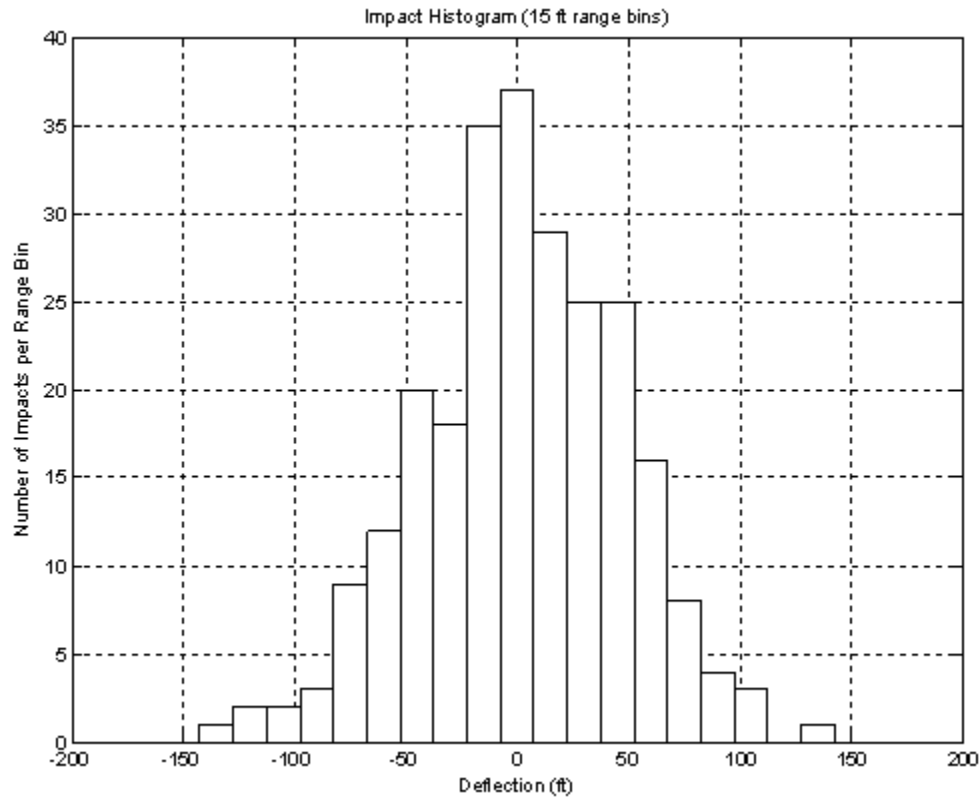


Figure 2 Impact Histogram

This data can also be used to calculate the probability that a bomb will land within a certain cross-range. For example, the bin from 37.5 to 52.5 ft contains 25 impacts. Therefore, knowing that 250 bombs were dropped, the probability that a bomb dropped will be in the range of 37.5 to 52.5 is $25/250$ or 10%. This technique is rather cumbersome however, so the histogram is replaced by a continuous expression known as a probability density function (PDF). [1]

1. Univariate Normal Distribution

a. Univariate Normal PDF

There are many different PDF's for different types of systems. The histogram in Figure 2 above resembles a common bell curve. This bell shaped curve is known as a normal distribution and represents many physical systems including many aspects of weapon delivery accuracy. As the number of samples is increased it can be shown that the analysis of the histogram above becomes a better approximation of the continuous expression for the univariate normal PDF given by:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} * e^{\left[\frac{-(x-\mu)^2}{2\sigma^2}\right]} \quad (2)$$

In Eq. (2), μ is the mean value of x and σ is the standard deviation. Approximately 68% of the datapoints for a normal distribution will be contained within $\pm\sigma$. [1]

b. Univariate Normal Cumulative Density Function (CDF)

The univariate normal cumulative density function is defined as:

$$F(x) = \int_{-\infty}^x \frac{1}{\sigma\sqrt{2\pi}} * e^{\left[\frac{-(x-\mu)^2}{2\sigma^2}\right]} dx \quad (3)$$

The CDF can be thought of as the probability that a given sample will lie in the range from $-\infty$ to some value X . Because this integral cannot be integrated analytically table references are commonly used. Standardization of the Eq (3) is used to allow for one table to be used for the various combinations of μ and σ . The transformation variable z in Eq (4) is used.

$$z = \frac{x - \mu}{\sigma} \quad (4)$$

Following substitution into Eq (3) and understanding that the standardized PDF has a zero mean and a standard deviation of 1 yields Eq (5). [1]

$$F(z) = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} * e^{\left[\frac{-z^2}{2}\right]} dz \quad (5)$$

If tables are unavailable, integration of Eq (3) can also be performed using MATLAB as shown in Table 1. For example, return to the previous example for the probability that a sample lies within the range of 37.5 to 52.5. Using Eq. (2) for the PDF and the MATLAB Symbolic Toolbox the value can be evaluated numerically. A known value of $\sigma = 50$ will be used for this evaluation. The random data generated for the 250 bomb impact sample was also based on $\sigma = 50$.

Table 1 MATLAB Code For Integrating the Non-standard Normal PDF

```
%-----
MATLAB CDF CODE:
% Input Mean
mu=0

% Input sigma
s=50

% Input Start of Bin
a=37.5

% Input End of Bin
b=52.5

% Input Total Number of Bombs
n=250

% Creates Symbolic univariate normal PDF
syms x
normpdf=1/(s*sqrt(2*pi))*exp(-((x-mu).^2)/(2*s^2))

% Numerically Calculates PDF Integral between specified values a and b
Prob=int(normpdf,a,b)
Prob=double(Prob)
numBombs=Prob*n
%-----
```

The above code results in a probability of 0.0798 that a given impact is contained within the range 37.5 to 52.5. Multiplying this value by the total number of bombs dropped results in the number of bombs contained in this range bin equal to

approximately 20 bombs. These values are close to the previously calculated probability of 0.1 with 25 bombs. If the number of bombs used to create the histogram is increased the histogram derived value will approach the numerically calculated 0.08. For example, with 50,000 bombs the histogram looks like Figure 3:

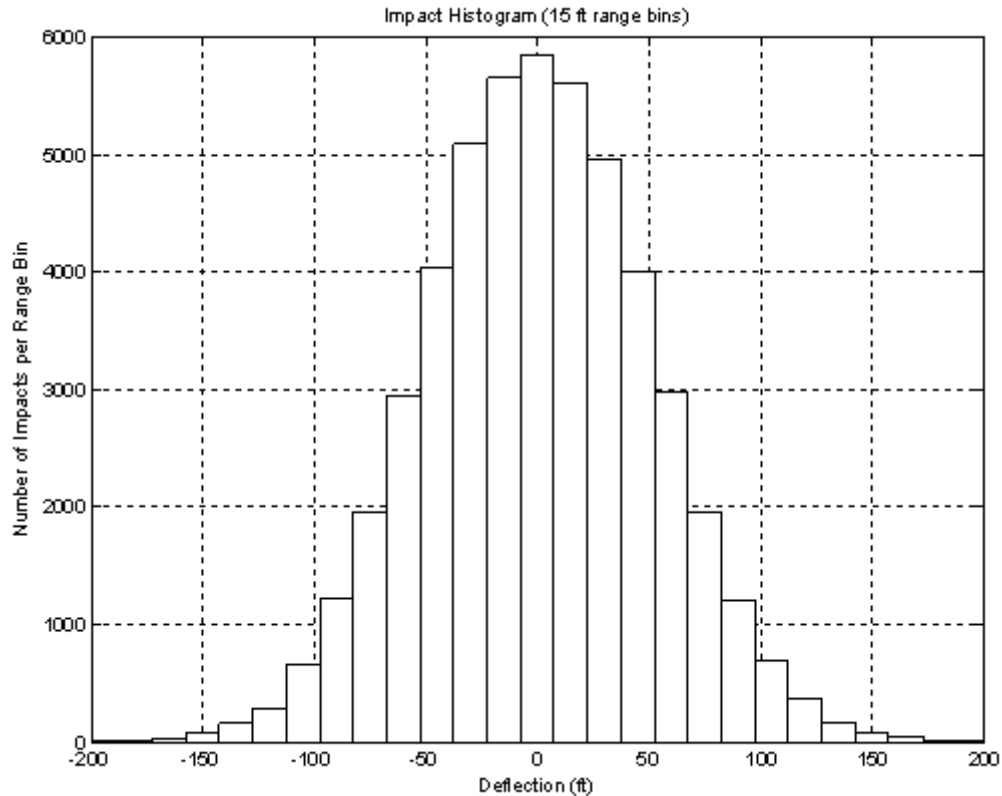


Figure 3 50,000 Samples Impact Histogram

Repeating the above MATLAB analysis with 50,000 bombs results in the same probability of 0.0798 (this is not dependent on number of samples) and 3,988 samples in the 37.5 to 52.5 range. The value displayed on the histogram of randomly generated samples contained in the deflection bin from 37.5 to 52.5 is 4,004 impacts. Therefore, the histogram value of 4004/50000 results in a probability of 0.0801. This is very good agreement and demonstrates the importance of convergence when modeling statistical data. Convergence is discussed more in Chapter II.

2. Bivariate Normal Distribution

a. Bivariate Normal PDF

The bivariate normal distribution can be thought of as the combination of two independent univariate normal distributions. While the univariate distribution will provide the probability that a bomb may fall within a certain one dimensional bin, the bivariate normal distribution can be used to determine the probability that a bomb will fall within a certain area. Take the bombing example and imagining that both deflection and range univariate normal distributions are used to determine the probability that a weapon will fall within a certain deflection and range distance from the DMPI. The bivariate normal PDF is shown below in Eq. (6)

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\left[\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right]} \quad (6)$$

b. Bivariate Normal CDF

The bivariate normal CDF is show below in Eq. (7)

$$F(X, Y) = \int_{x=-\infty}^{x=X} \int_{y=-\infty}^{y=Y} \frac{1}{2\pi\sigma_x\sigma_y} e^{-\left[\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right]} dx dy \quad (7)$$

3. Circular Normal and Rayleigh Distributions

a. Circular Normal PDF

The circular normal distribution is a bivariate normal distribution with zero means and equal standard deviations for x and y. This causes Eq. (6) to reduce to Eq. (8)

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left[\frac{x^2+y^2}{2\sigma^2}\right]} \quad (8)$$

b. Rayleigh PDF

The Rayleigh PDF is defined as the distribution of the value r defined as

$$r^2 = x^2 + y^2 \quad (9)$$

After some manipulation, this results in the Rayleigh PDF shown in Eq. (10)

$$f(r) = \frac{r}{\sigma^2} e^{\left[\frac{-r^2}{2\sigma^2}\right]} \quad (10)$$

c. Rayleigh CDF

The Rayleigh PDF can be integrated analytically resulting in the Rayleigh CDF in Eq. (11)

$$F(R) = 1 - e^{\left[\frac{-R^2}{2\sigma^2}\right]} \quad (11)$$

B. ERROR TYPES

There are two main types of error to be defined when discussing unguided weapon deliveries. The error can be broken down into ballistic dispersion error and aiming error.

1. Ballistic Dispersion

Ballistic dispersion error is defined as the error in the weapon delivery caused by physical inconsistencies between individual weapons (weight, center of gravity, fin shape/angle bias, surface deviations, etc.). These inconsistencies cause each weapon's ballistic trajectory to be slightly different. The random physical inconsistencies typically result in weapon delivery behavior that can be represented by a normal distribution. [1]

2. Aiming Error

Aiming error is the difference between the actual target location and the weapon system aim point. This error is also considered to be normally distributed.

C. ACCURACY

Numerous parameters can be used to characterize the accuracy of a weapon system. Some of the most useful parameters are circular error probable (CEP), range error probable (REP), and deflection error probable (DEP).

1. Circular Error Probable (CEP)

The CEP is defined as the radius of a circle (centered on the DMPI) that contains 50% of the bomb impacts (see Figure 4).

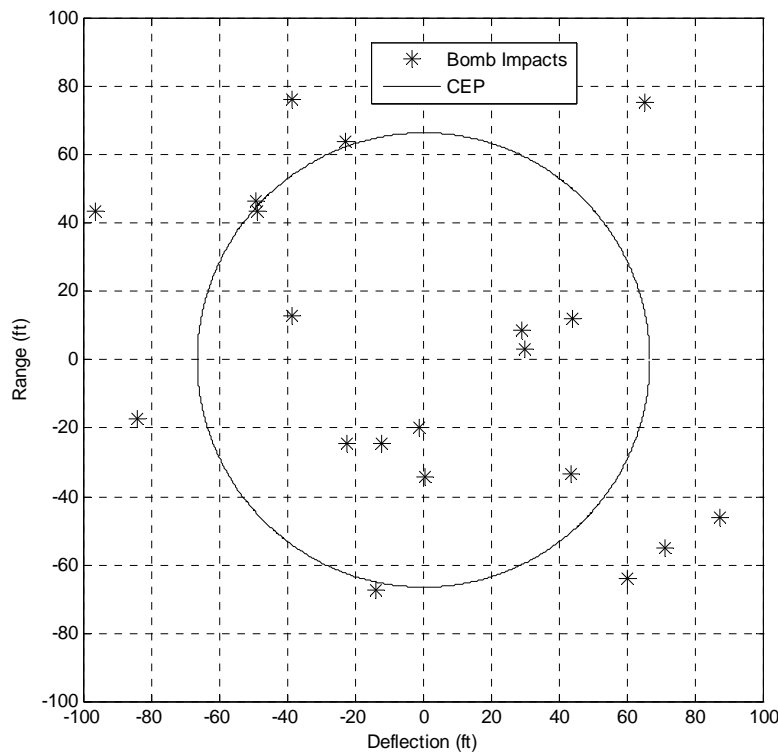


Figure 4 Circular Error Probable

2. Range Error Probable (REP) and Deflection Error Probable (DEP)

The REP is defined as a length of half of a range-bin centered at the DMPI that contains half of the impacts in the range direction. The DEP is defined as a length of half of a range-bin centered at the DMPI that contains half of the impacts in the deflection direction (see Figure 5).

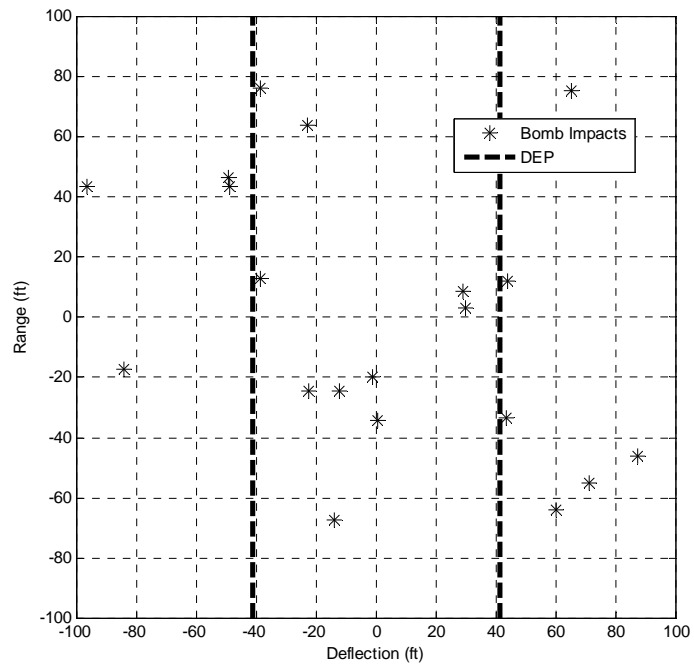
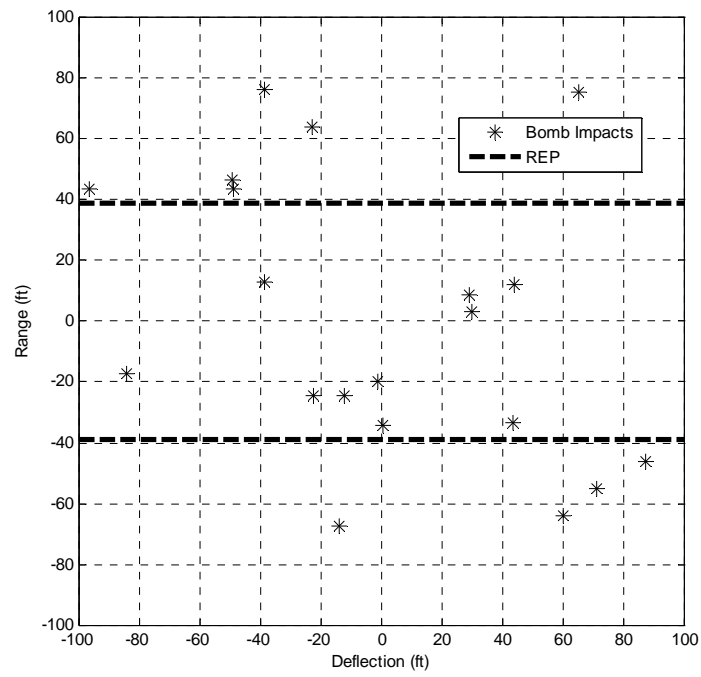


Figure 5 Range Error Probable and Deflection Error Probable

3. Relationship of CEP to REP/DEP

The following equations describe the relationship between REP/DEP and standard deviation in the range and deflection directions.

$$REP = 0.6745\sigma_x \quad (12)$$

$$DEP = 0.6745\sigma_y \quad (13)$$

It is also observed that if the data distribution has a zero mean and is assumed to be circular (standard deviations in the range and deflection direction are equal) the following relationships between σ , CEP, and REP/DEP hold. [1]

$$CEP = 1.1774\sigma \quad (14)$$

$$CEP = 1.7456REP = 1.7456DEP \quad (15)$$

D. COMBINING ERROR TYPES

Once the statistics of the accuracy of a weapon are understood it is then important to define the probability that a given weapon, or group of similar weapons, will damage a specific target. This is the study of weapon effectiveness and is a function of the weapon, target, and scenario. This analysis has the potential to result in a very complex calculation. For the purposes of discussions herein it will be assumed that the weapon accuracy statistics (σ , CEP, REP/DEP) for the scenario are provided and the weapon area of effectiveness is also provided based on known weapon/target/scenario characteristics. An individual weapon is defined to have hit the target if its impact is close enough for the area of effectiveness to enclose the target. Conversely, the area of effectiveness can be centered on the target and a hit can be defined as a weapon impacting inside the area of effectiveness. Both of these hit definitions are equivalent representations. Having these parameters will allow for comparison of analytical approximations and MATLAB simulation results for two specific scenarios: single round per occasion and multiple round salvos per occasion. An occasion is defined as an event for which aiming error is considered constant. For example, an aircraft delivering two bombs (at the same target) on one pass is one occasion. An example of two occasions is an aircraft delivering two bombs on two passes, one bomb per pass.

1. Single Round Scenario

As discussed in section 1.B there are two main error types of interest: aiming error (σ_{aim}) and ballistic dispersion (σ_{bd}). Take any given round and assume these two errors are known. To properly simulate the weapon impact location both errors need to be accounted for as shown in Figure 6.

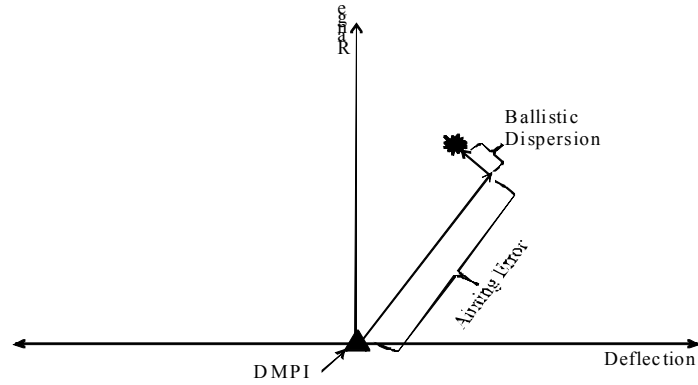


Figure 6 Single Round: Aiming Error and Ballistic Dispersion

This is an example of one occasion of a single round delivery. A four occasion single round delivery can be seen in Figure 7.

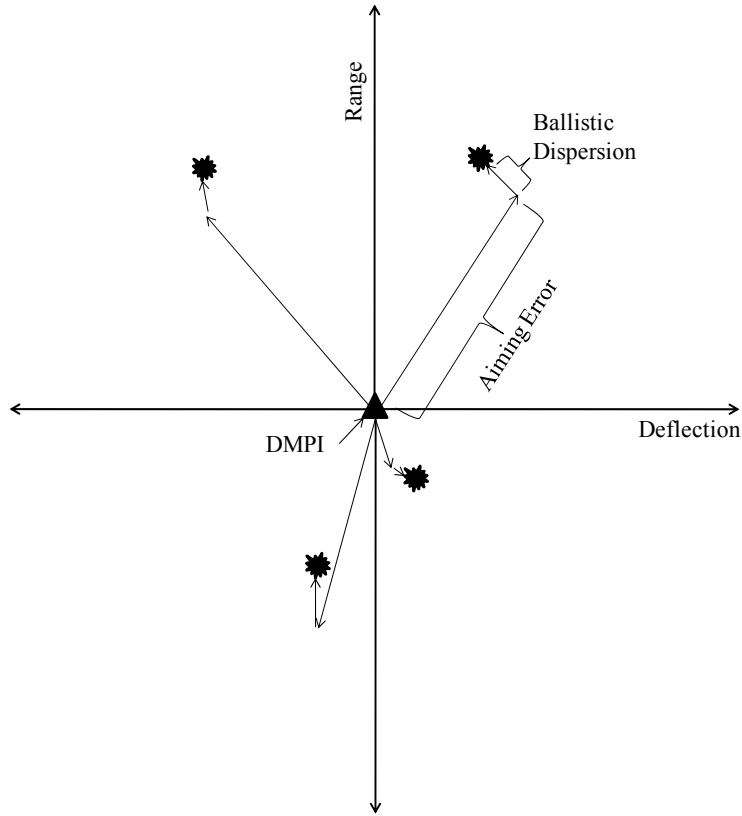


Figure 7 Single Round Scenario: Four Occasions

For a typical scenario, as shown in Figure 6 and Figure 7, it is common that σ_{aim} and σ_{bd} are not equal. Usually, the aiming error is a more significant error contribution than the ballistic dispersion.

2. Salvo Scenario

A salvo, as one might expect, is defined as multiple rounds per occasion. The salvo is used to increase the probability of damage to the target. An example of a four occasion scenario with 5 bombs per salvo is shown in Figure 8.

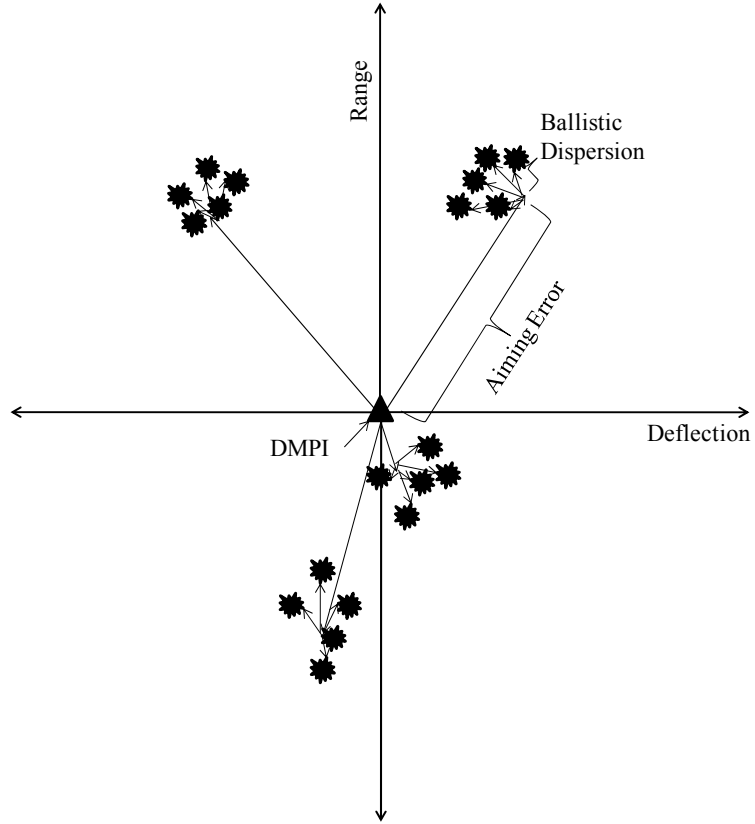


Figure 8 Salvo Scenario: Four Occasions; Five Bombs/Salvo

It is interesting to note that these two scenarios have significantly different calculations to determine the probability of hit. For the single round scenario the two error types, ballistic dispersion and aiming, can be considered independent for each weapon. For the salvo scenario this is clearly not the case as the aiming error for one salvo biases the results equally for all bombs of a given salvo. This results in a more complicated analysis to properly calculate the probability of at least one hit on the target given a salvo scenario. Examples of two approximations to salvo effectiveness are shown in Eq. (16) and Eq. (17).

$$p_n(n) = \frac{1}{c} \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} \left(\frac{\frac{R_T^2}{\sigma_x^2}}{2c + \frac{R_T^2}{\sigma_x^2}} \right)^{i-1} \left[1 - \exp \left(- \frac{\frac{R_T^2}{\sigma_x^2}}{\frac{2c + \frac{R_T^2}{\sigma_x^2}}{ci} + \frac{2\sigma_\mu^2}{\sigma_x^2}} \right) \right] \quad (16)$$

$$p_n(n) = \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} \left[\frac{R_T}{\sigma_x^2 \left(2 + \frac{R_T^2}{\sigma_x^2} \right)} \right]^{i-1} \left[\frac{\frac{R_T}{\sigma_x^2}}{2 + \frac{R_T}{\sigma_x^2} + i \left(\frac{2\sigma_\mu^2}{\sigma_x^2} \right)} \right] \quad (17)$$

Where:

$p_n(n)$ = Probability at least one round hits the target

n = Number of Rounds/Salvo

R_T = Target Radius (also can be thought of as the effective weapon radius)

σ_μ = Aiming Error Standard Deviation

σ_x = Ballistic Dispersion Standard Deviation

c = Adjustment Factor (Typically 0.9 to 1.0) [2]

Monte-Carlo simulations as outlined in the following section can also be used to evaluate the salvo scenario effectiveness. Chapter III will deal exclusively with accuracy and probability of hit calculations and simulations for both the single round and salvo scenarios.

3. Simulation Implementation

The Monte-Carlo simulation used to generate salvo scenario results are performed as shown in Figure 9.

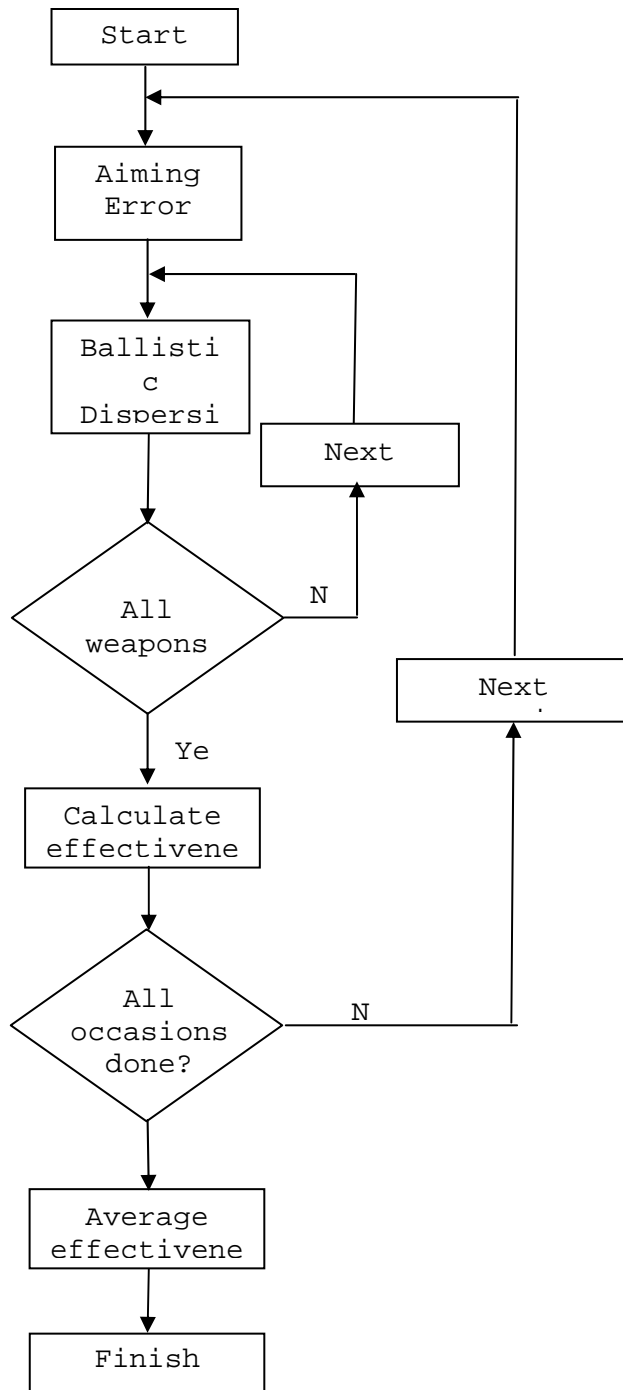


Figure 9 Salvo Effectiveness Simulation Flowchart [From 3]

The process shown in Figure 9 can be easily implemented in MATLAB allowing for potentially more accurate results than the analytical approximations to be discussed in Chapter III.

II. TEST DATA CHARACTERISTICS

When comparing statistical datasets with analytical solutions/approximations it is critical to monitor convergence. Convergence, for the purposes herein, will be handled using the number of samples defined for a given simulation. The simulation result is compared to an appropriate analytical expression if available. By studying the comparison of multiple simulation runs and analytical results convergence can be observed. If the dataset is not suitably converged the number of samples is increased and the simulation is repeated. This process is continued until suitable convergence is achieved.

A. UNIVARIATE NORMAL

The REP of a univariate normal distribution is shown in Eq. (18).

$$REP = 0.6745\sigma_x \quad (18)$$

This equation is the analytical representation of the CEP. To properly compare this analytical representation to simulation results it is important to monitor the convergence of the data set. For example, a given dataset has a known $\sigma = 50\text{ ft}$ which will result in a REP of 33.725 ft. Using a Monte-Carlo simulation to generate a normal dataset and extract the REP for various numbers of samples results in Table 2.

Table 2 REP Convergence Data

Number of Bombs Simulated	REP (ft)		
	Run 1	Run 2	Run 3
10	64.500	38.363	42.723
10^2	33.105	28.292	26.624
10^3	31.955	33.616	35.027
10^4	33.369	33.794	33.792
10^5	33.655	33.535	33.430
10^6	33.716	33.732	33.706
10^7	33.742	33.720	33.725

It can be seen that even with 107 samples the simulation has not converged to the exact value for REP calculated from Eq. (18). However, the practice of weapon engineering rarely demands this level of precision allowing for an acceptable number of samples to be used to generate appropriately converged Monte-Carlo simulation results.

B. RAYLEIGH

The Rayleigh distribution is given by Eq. (19)

$$f(r) = \frac{r}{\sigma^2} e^{\left[\frac{-r^2}{2\sigma^2}\right]} \quad (19)$$

The standard deviation in the above equation is the common standard deviation in the x and y directions (the Rayleigh distribution requiring a circular normal distribution in the x and y directions). Again, take a known $\sigma = 50 \text{ ft}$ (common in the x and y directions) for a circular normal dataset. For circular normal distributions the CEP can be calculated using Eq. (14) which yields a CEP of 58.870 ft. Running a Monte-Carlo simulation to generate a circular normal dataset for various numbers of samples and extracting CEP yields Table 3.

Table 3 CEP Convergence Data

Number of Bombs Simulated	CEP (ft)		
	Run 1	Run 2	Run 3
10	62.485	70.225	72.979
10^2	60.084	55.831	57.434
10^3	58.398	58.803	58.136
10^4	58.978	59.727	58.479
10^5	58.918	58.948	58.823
10^6	58.881	58.883	58.879
10^7	58.868	58.869	58.865

C. SALVO FORMULA

The salvo formula calculations are significantly more complex than the simple examples given for REP and CEP. However, the process remains the same. Using Eq. (20):

$$p_n(n) = \frac{1}{c} \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} \left(\frac{\frac{R_T^2}{\sigma_x^2}}{2c + \frac{R_T^2}{\sigma_x^2}} \right)^{i-1} \left[1 - \exp \left(- \frac{\frac{R_T^2}{\sigma_x^2}}{\frac{2c + \frac{R_T^2}{\sigma_x^2}}{ci} + \frac{2\sigma_\mu^2}{\sigma_x^2}} \right) \right] \quad (20)$$

With:

$$n = 7$$

$$R_T = 200 \text{ ft}$$

$$\sigma_\mu = 150 \text{ ft}$$

$$\sigma_x = 50 \text{ ft}$$

$$c = 1.0$$

Results in a $p_n(n) = 0.762$.

Again, a Monte-Carlo simulation can be performed varying the number of occasions to achieve convergence. The results of this simulation are shown on Table 4.

Table 4 Salvo Formula Convergence Data

Number of Occasions Simulated	$p_n(n)$		
	Run 1	Run 2	Run 3
10	0.600	0.800	0.600
10^2	0.780	0.770	0.720
10^3	0.788	0.749	0.787
10^4	0.777	0.774	0.778
10^5	0.775	0.772	0.774
10^6	0.773	0.773	0.773
10^7	0.773	0.773	0.773

It is important to remember that Eq. (20) is an analytical approximation for the salvo effectiveness. Using the Monte-Carlo simulation therefore can yield more accurate results than the analytical approximation. The approximation still gives reasonably close results and is only 0.011 from the converged simulation value of 0.773. However, this demonstrates the potential value of the Monte-Carlo simulation process by taking advantage of modern computing power to run enough iterations to calculate a more accurate result than an approximation can provide.

III. MAINTAINING AIMING ERROR AND BALLISTIC DISPERSION AS SEPARATE PARAMETERS

In the study of statistics it is often suitable to combine multiple probabilities to simplify the analysis. One possible combination is taking the root-sum-square (RSS) of two independent standard deviations of similar distributions that are used to define the behavior of a total population. However, care must be taken if performing this operation when addressing the weapon delivery standard deviations of aiming error and ballistic dispersion. When attempting to calculate accuracy parameters (CEP or REP/DEP) the RSS simplification is appropriate. However, when calculating weapon effectiveness for a salvo of munitions the aiming error and ballistic dispersion must be maintained as separate parameters. These results are demonstrated through the use of simulation in the following sections.

A. ACCURACY CALCULATIONS

The accuracy calculations performed below are based on a simulation using an algorithm as outlined in Figure 10. This algorithm assumes a circular normal distribution with zero mean for both error types. It is also possible to easily modify the algorithm to perform noncircular calculations. This is one of the significant advantages to using simulation practices versus analytical approaches. Many analytical approaches require substantial mathematical manipulation and potential approximation or numerical solutions to yield useful results. By creating the proper simulation routine, a very complex weapon accuracy model can be evaluated in a very similar fashion to the simple model pictured here.

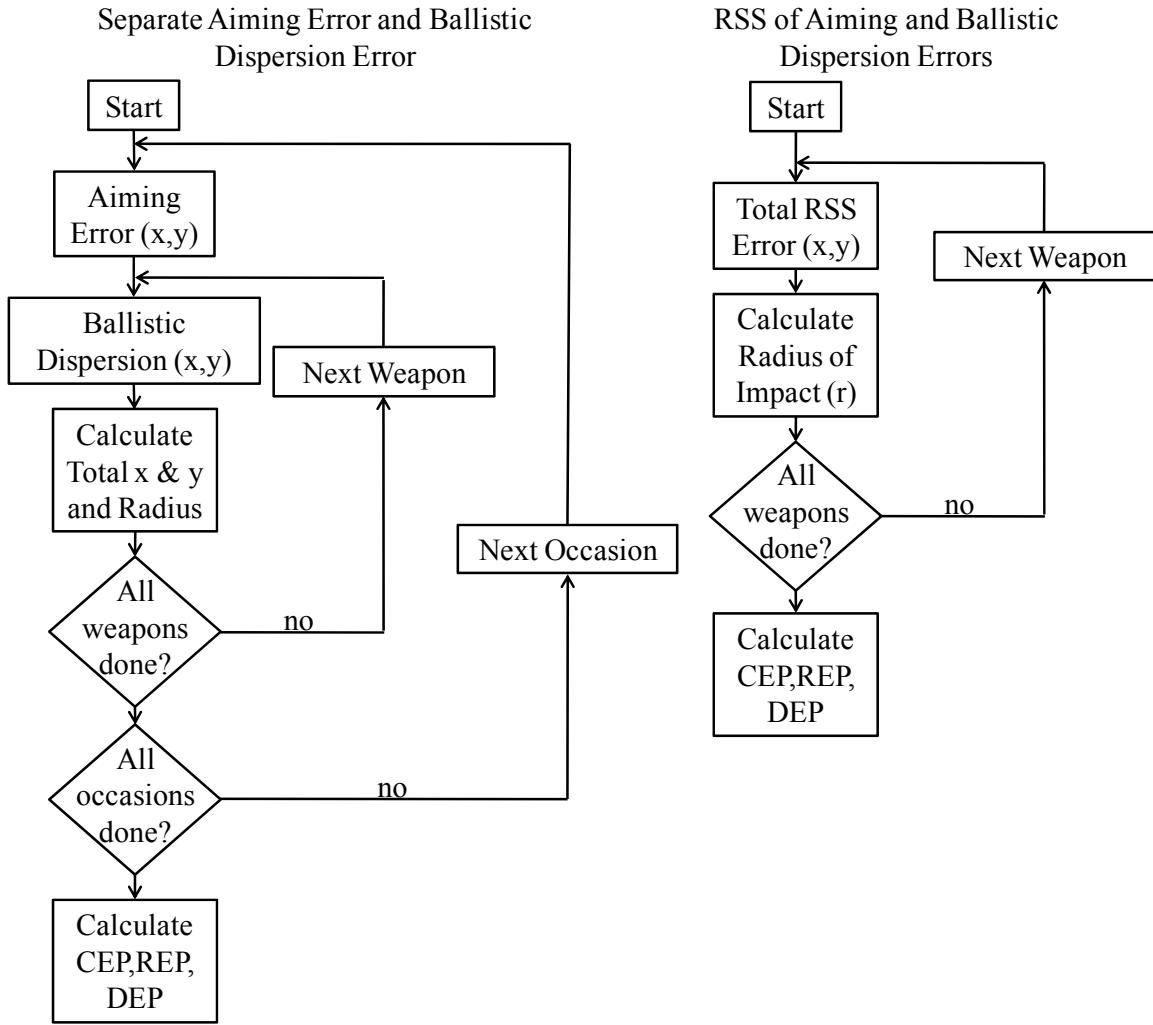


Figure 10 Flowchart For Accuracy Simulations

1. Single Round Scenario

The single round accuracy results for various ratios of aiming error and ballistic dispersion show good correlation between the calculated values for CEP, REP, and DEP using both algorithms (separate errors vs. RSS error). Simulations were performed using 10^6 occasions as outlined in Figure 10. This number of occasions allowed for proper convergence of the results. The results are shown below on Table 5. Results can also be compared to the values calculated from Eq. (12), Eq. (13) and Eq. (14).

Table 5 Separate vs. RSS Errors for Accuracy Calculations (1 bomb per salvo)

ERRORS	CEP		REP		DEP	
	Separate	RSS	Separate	RSS	Separate	RSS
$\sigma_{\text{aiming}}=50$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=50.3$	59.19	59.17	33.91	33.90	33.90	33.91
$\sigma_{\text{aiming}}=5$ $\sigma_{\text{bd}}=50$ $\sigma_{\text{RSS}}=50.3$	59.15	59.15	33.89	33.88	33.87	33.90
$\sigma_{\text{aiming}}=500$ $\sigma_{\text{bd}}=500$ $\sigma_{\text{RSS}}=707.1$	833.0	832.6	477.0	476.9	477.3	476.8
$\sigma_{\text{aiming}}=1000$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=1000.01$	1,178	1,177	674.5	674.2	674.4	675.1
$\sigma_{\text{aiming}}=953.95$ $\sigma_{\text{bd}}=300$ $\sigma_{\text{RSS}}=1000.01$	1,177	1,178	674.4	674.4	674.7	674.5

It is clear from the above table that the aiming error and ballistic dispersion error can be root-sum-squared to simplify the calculations for the accuracy parameters. Additionally, calculating the RSS (or total) standard deviation from the aiming and ballistic dispersion standard deviations allows for the use of Eq. (12), Eq. (13) and Eq. (14) to easily calculate the accuracy parameters for the single round scenario.

2. Salvo Scenario

The salvo scenario analysis for accuracy calculations also uses the processes outlined in Figure 10. However, the weapon loops will now be used due to the salvo having a greater than one number of rounds per each occasion. Runs of 5, 10, 50 and 100 weapons per salvo were performed with the same standard deviations as used in the single round scenario analysis. These results from the salvo accuracy parameter analysis can be seen on Table 6 - Table 9.

Table 6 Separate vs. RSS Errors for Accuracy Calculations (5 bombs per salvo)

ERRORS	CEP		REP		DEP	
	Separate	RSS	Separate	RSS	Separate	RSS
$\sigma_{\text{aiming}}=50$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=50.3$	59.12	59.17	33.87	33.89	33.88	33.88
$\sigma_{\text{aiming}}=5$ $\sigma_{\text{bd}}=50$ $\sigma_{\text{RSS}}=50.3$	59.17	59.16	33.88	33.92	33.89	33.89
$\sigma_{\text{aiming}}=500$ $\sigma_{\text{bd}}=500$ $\sigma_{\text{RSS}}=707.1$	832.5	832.8	476.7	477.2	477.1	477.0
$\sigma_{\text{aiming}}=1000$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=1000.01$	1,178	1,178	673.6	675.1	674.8	674.8
$\sigma_{\text{aiming}}=953.95$ $\sigma_{\text{bd}}=300$ $\sigma_{\text{RSS}}=1000.01$	1,177	1,177	674.7	674.2	673.7	673.9

Table 7 Separate vs. RSS Errors for Accuracy Calculations (10 bombs per salvo)

ERRORS	CEP		REP		DEP	
	Separate	RSS	Separate	RSS	Separate	RSS
$\sigma_{\text{aiming}}=50$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=50.3$	59.11	59.18	33.86	33.90	33.90	33.91
$\sigma_{\text{aiming}}=5$ $\sigma_{\text{bd}}=50$ $\sigma_{\text{RSS}}=50.3$	59.19	59.15	33.9	33.89	33.90	33.90
$\sigma_{\text{aiming}}=500$ $\sigma_{\text{bd}}=500$ $\sigma_{\text{RSS}}=707.1$	832.7	832.6	477.1	477.0	477.1	477.1
$\sigma_{\text{aiming}}=1000$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=1000.01$	1,177	1,178	673.3	674.5	675.3	674.5
$\sigma_{\text{aiming}}=953.95$ $\sigma_{\text{bd}}=300$ $\sigma_{\text{RSS}}=1000.01$	1,177	1,178	674.5	674.5	674.2	674.7

Table 8 Separate vs. RSS Errors for Accuracy Calculations (50 bombs per salvo)

ERRORS	CEP		REP		DEP	
	Separate	RSS	Separate	RSS	Separate	RSS
$\sigma_{\text{aiming}}=50$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=50.3$	59.15	59.16	33.86	33.89	33.84	33.89
$\sigma_{\text{aiming}}=5$ $\sigma_{\text{bd}}=50$ $\sigma_{\text{RSS}}=50.3$	59.16	59.16	33.89	33.89	33.89	33.89
$\sigma_{\text{aiming}}=500$ $\sigma_{\text{bd}}=500$ $\sigma_{\text{RSS}}=707.1$	832.8	832.4	477.4	476.9	476.9	476.9
$\sigma_{\text{aiming}}=1000$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=1000.01$	1,179	1,177	674.2	674.4	676.3	674.4
$\sigma_{\text{aiming}}=953.95$ $\sigma_{\text{bd}}=300$ $\sigma_{\text{RSS}}=1000.01$	1,177	1,178	675.3	674.5	674.1	674.6

Table 9 Separate vs. RSS Errors for Accuracy Calculations (100 bombs per salvo)

ERRORS	CEP		REP		DEP	
	Separate	RSS	Separate	RSS	Separate	RSS
$\sigma_{\text{aiming}}=50$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=50.3$	59.20	59.17	33.91	33.89	33.91	33.89
$\sigma_{\text{aiming}}=5$ $\sigma_{\text{bd}}=50$ $\sigma_{\text{RSS}}=50.3$	59.17	59.16	33.90	33.90	33.90	33.89
$\sigma_{\text{aiming}}=500$ $\sigma_{\text{bd}}=500$ $\sigma_{\text{RSS}}=707.1$	832.6	832.5	476.9	476.9	477.1	476.9
$\sigma_{\text{aiming}}=1000$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=1000.01$	1,177	1,177	673.7	674.5	673.6	674.5
$\sigma_{\text{aiming}}=953.95$ $\sigma_{\text{bd}}=300$ $\sigma_{\text{RSS}}=1000.01$	1,178	1,177	674.8	674.5	674.7	674.5

It is interesting to note that on Table 5 thru Table 9 the two methods for calculating the accuracy parameters both show similar results for the various scenarios regardless of the number of weapons per salvo. This demonstrates that for accuracy parameter calculation it is appropriate to RSS the aiming error and ballistic dispersion error standard deviations for both single round and salvo scenarios.

B. WEAPON EFFECTIVENESS CALCULATIONS

For weapon effectiveness simulations the procedure outlined in Figure 11 was used. This algorithm can be easily modified to allow for a more complex weapon/target lethal area than is available with analytical approaches.

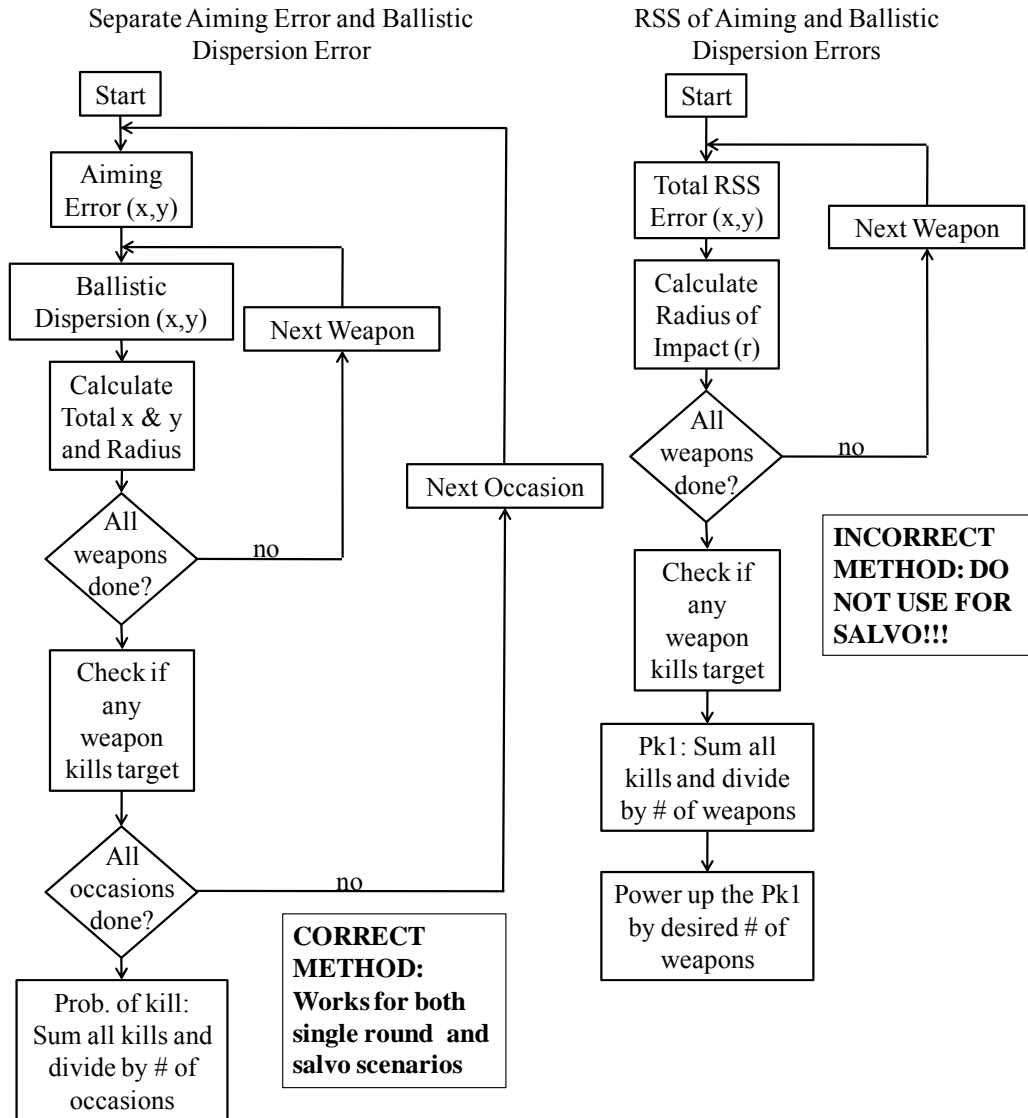


Figure 11 Flow Chart for Weapon Effectiveness Simulations

1. Single Round Scenario

The single round scenario will provide reasonable results using both of the methods outlined in Figure 11. The Mean Area of Effectiveness was set to 7854 ft² corresponding to a lethal radius of 50 ft. Simulation results can be seen in Table 10.

An analytical approach can also be used to check the single round scenario algorithms. Using the CDF for the Rayleigh distribution will yield the probability that a given round will fall within the radius R. If R is set to the lethal radius and the standard deviation is set to the RSS value of the given aiming and ballistic dispersion errors the CDF value will equal the probability that a given round will land within the lethal radius of the weapon. This is precisely the weapon effectiveness parameter of interest for the single round scenario. Knowing that the Rayleigh CDF is:

$$F(R) = 1 - e^{\left[\frac{-R^2}{2\sigma^2}\right]} \quad (21)$$

Where: R = Weapon Lethal Radius = 50 ft

$$\sigma = \sqrt{\sigma_{\text{aiming}}^2 + \sigma_{\text{bd}}^2}$$

Table 10 Separate vs. RSS Errors for Weapon Effectiveness (1 bomb per salvo)

ERRORS	p _n (1)		Eq (21)
	Separate	RSS	Analytical Sol.
$\sigma_{\text{aiming}}=50$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=50.3$	0.39	0.39	0.39
$\sigma_{\text{aiming}}=5$ $\sigma_{\text{bd}}=50$ $\sigma_{\text{RSS}}=50.3$	0.39	0.39	0.39
$\sigma_{\text{aiming}}=500$ $\sigma_{\text{bd}}=500$ $\sigma_{\text{RSS}}=707.1$	0.0025	0.0025	0.0025
$\sigma_{\text{aiming}}=1000$ $\sigma_{\text{bd}}=5$ $\sigma_{\text{RSS}}=1000.01$	0.0012	0.0012	0.0012
$\sigma_{\text{aiming}}=953.95$ $\sigma_{\text{bd}}=300$ $\sigma_{\text{RSS}}=1000.01$	0.0012	0.0012	0.0012

It is clear from the results in Table 10 that either method is suitable for determining the weapon system effectiveness for a single round scenario.

2. Salvo Scenario

The salvo scenario analysis for accuracy calculations uses the processes outlined in Figure 11. However, the weapon loops will now be used due to the salvo having a greater than one number of rounds per occasion. Runs of 5, 10, 50 and 100 weapons per salvo were performed with the same standard deviations as used in the single round scenario analysis. These results from the salvo weapon effectiveness analysis can be seen on Table 11.

Table 11 Separate vs. RSS Errors for Weapon Effectiveness (5,10,50,100 bombs per salvo)

ERRORS	$p_n(5)$		$p_n(10)$		$p_n(50)$		$p_n(100)$	
	Separate	RSS	Separate	RSS	Separate	RSS	Separate	RSS
$\sigma_{aiming}=50$ $\sigma_{bd}=5$ $\sigma_{RSS}=50.3$	0.46	0.92	0.48	1.00	0.52	1.00	0.54	1.00
$\sigma_{aiming}=5$ $\sigma_{bd}=50$ $\sigma_{RSS}=50.3$	0.92	0.92	0.99	0.99	1.00	1.00	1.00	1.00
$\sigma_{aiming}=500$ $\sigma_{bd}=500$ $\sigma_{RSS}=707.1$	0.012	0.012	0.025	0.025	0.12	0.12	0.21	0.23
$\sigma_{aiming}=1000$ $\sigma_{bd}=5$ $\sigma_{RSS}=1000.01$	0.0016	0.0062	0.0016	0.012	0.0021	0.061	0.0018	0.12
$\sigma_{aiming}=953.95$ $\sigma_{bd}=300$ $\sigma_{RSS}=1000.01$	0.0062	0.0062	0.012	0.012	0.053	0.059	0.092	0.11

Table 11 demonstrates the fundamental reason that the aiming error and ballistic dispersion error standard deviations must be treated separately for the purposes of weapon effectiveness calculations. The individual weapons of a given salvo are all influenced by the same aiming error. This results in the weapon impact location for each weapon in the salvo being dependent on a constant aiming error. For a given salvo to be considered successful in killing the target at least one bomb of that salvo must impact within the lethal area. The single round separate error scenario however, will count each weapon inside the lethal area as a kill and consequently result in a similar effectiveness

parameter as that of the RSS weapon effectiveness. To check this assumption, an algorithm can be setup to count each bomb that lands inside the lethal area during the separate algorithm operation. Count all the bombs that fall inside the lethal area and divide by the total number of bombs dropped to yield the chance of a given bomb landing inside the lethal area. Then, this number must be powered up to provide the incorrect total salvo weapon effectiveness that will correspond to the incorrect value given by the RSS algorithm. This fundamental difference results in the RSS algorithm providing incorrectly optimistic weapon effectiveness for weapon salvos. For the scenarios where the aiming error is significantly large than the ballistic dispersion error the separate vs. RSS effectiveness values are significantly different. This error is a function of σ_{aiming} , σ_{bd} , and lethal radius. Clearly, for proper calculation of weapon effectiveness, the aiming error and ballistic dispersion errors must be kept separate and the procedure on the left side of Figure 11 should be used.

IV. SALVO EFFECTIVENESS CALCULATIONS: SIMULATION VS. ANALYTICAL APPROXIMATIONS

It was observed in Chapter II that the simulation for salvo weapon effectiveness could yield more accurate results than the analytical approximations historically used. Chapter IV will provide additional detail regarding these approximation deficiencies.

A. CIRCULAR TARGET

The following charts compare simulation runs with Eq. (16) and Eq. (17) salvo formula approximations. Two sets of errors were evaluated. One set with aiming error of 50 and ballistic dispersion of 5 (as investigated previously). The other set increased the ballistic dispersion and uses aiming error of 50 and ballistic dispersion of 25. Weapon effectiveness for both of these error sets was calculated for multiple weapon lethal areas (5, 10, 20, 30, 40, 50, 60). The results are shown below on Table 12 thru Table 25.

Table 12 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=5$; Lethal Radius 5)

Lethal Radius	5				
#/Salvo	Sim Circle	Eq. 16	Eq. 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.005	0.005	0.005	0.000	0.000
10	0.025	0.027	0.027	-0.002	-0.002
20	0.033	0.037	0.037	-0.003	-0.003
30	0.038	0.042	0.042	-0.004	-0.004
40	0.041	0.046	0.046	-0.005	-0.005
50	0.044	0.049	0.050	-0.006	-0.006
60	0.046	0.052	0.052	-0.006	-0.006
70	0.048	0.054	0.054	-0.006	-0.006
80	0.049	0.056	0.056	-0.008	-0.008
90	0.050	0.058	0.058	-0.007	-0.008
100	0.052	0.059	0.059	-0.007	-0.007

Table 13 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=5$; Lethal Radius 10)

Lethal Radius	10				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.020	0.019	0.019	0.001	0.000
10	0.055	0.072	0.072	-0.016	-0.017
20	0.066	0.090	0.091	-0.024	-0.024
30	0.073	0.100	0.101	-0.028	-0.029
40	0.078	0.108	0.109	-0.030	-0.031
50	0.080	0.114	0.115	-0.034	-0.035
60	0.082	0.119	0.120	-0.037	-0.038
70	0.085	0.120	0.124	-0.034	-0.038
80	0.087	-3.42E-01	0.123	0.430	-0.035
90	0.090	-5.73E+01	-6.13E-01	5.74E+01	7.04E-01
100	0.090	-6.07E+03	-1.33E+02	6.07E+03	1.33E+02

Table 14 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=5$; Lethal Radius 20)

Lethal Radius	20				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.076	0.071	0.073	0.005	0.003
10	0.138	0.211	0.219	-0.074	-0.081
20	0.153	0.256	0.264	-0.103	-0.112
30	0.163	0.281	0.290	-0.118	-0.127
40	0.169	0.298	0.308	-0.130	-0.139
50	0.173	0.311	0.322	-0.139	-0.149
60	0.176	0.298	0.330	-0.122	-0.154
70	0.179	-1.36E+01	-2.46E+00	1.38E+01	2.64E+00
80	0.181	-3.08E+03	-6.13E+02	3.08E+03	6.14E+02
90	0.182	-5.17E+05	-8.60E+05	5.17E+05	8.60E+05
100	0.186	5.44E+07	-2.31E+08	-5.44E+07	2.31E+08

Table 15 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=5$; Lethal Radius 30)

Lethal Radius	30				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.163	0.140	0.151	0.022	0.011
10	0.244	0.380	0.406	-0.136	-0.162
20	0.262	0.448	0.476	-0.186	-0.214
30	0.273	0.485	0.514	-0.212	-0.241
40	0.281	0.510	0.540	-0.229	-0.259
50	0.285	0.528	0.559	-0.243	-0.274
60	0.288	0.592	0.592	-0.304	-0.304
70	0.293	-5.44E+01	-6.82E+01	5.47E+01	6.85E+01
80	0.296	1.51E+04	-8.73E+03	-1.51E+04	8.73E+03
90	0.298	-4.75E+07	-1.47E+07	4.75E+07	1.47E+07
100	0.301	-2.88E+10	-1.01E+10	2.88E+10	1.01E+10

Table 16 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=5$; Lethal Radius 40)

Lethal Radius	40				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.271	0.214	0.241	0.057	0.030
10	0.364	0.535	0.587	-0.172	-0.223
20	0.381	0.615	0.668	-0.234	-0.286
30	0.395	0.656	0.708	-0.261	-0.314
40	0.401	0.683	0.734	-0.282	-0.334
50	0.406	0.702	0.753	-0.296	-0.347
60	0.408	0.797	0.930	-0.389	-0.522
70	0.412	5.18E+01	-1.01E+02	-5.14E+01	1.02E+02
80	0.416	1.46E+05	-5.11E+04	-1.46E+05	5.11E+04
90	0.419	-1.18E+08	-6.46E+07	1.18E+08	6.46E+07
100	0.420	-1.81E+11	-6.06E+10	1.81E+11	6.06E+10

Table 17 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=5$; Lethal Radius 50)

Lethal Radius	50				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.390	0.282	0.331	0.108	0.059
10	0.485	0.658	0.733	-0.173	-0.248
20	0.503	0.739	0.809	-0.236	-0.307
30	0.513	0.778	0.844	-0.265	-0.331
40	0.519	0.802	0.865	-0.283	-0.346
50	0.524	0.819	0.880	-0.295	-0.356
60	0.529	0.790	1.19E+00	-0.261	-0.665
70	0.535	-1.87E+02	-3.79E+02	1.88E+02	3.80E+02
80	0.535	9.30E+04	1.51E+05	-9.30E+04	-1.51E+05
90	0.538	-3.42E+08	-1.76E+08	3.42E+08	1.76E+08
100	0.540	-1.72E+11	-8.75E+10	1.72E+11	8.75E+10

Table 18 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=5$; Lethal Radius 60)

Lethal Radius	60				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.510	0.340	0.416	0.169	0.093
10	0.597	0.748	0.838	-0.151	-0.241
20	0.613	0.824	0.899	-0.211	-0.287
30	0.622	0.858	0.924	-0.236	-0.302
40	0.630	0.878	0.938	-0.249	-0.308
50	0.636	0.892	0.949	-0.255	-0.313
60	0.639	1.75E+00	1.91E+00	-1.11E+00	-1.27E+00
70	0.642	4.83E+01	-1.91E+02	-4.77E+01	1.91E+02
80	0.645	4.77E+05	1.03E+04	-4.77E+05	-1.03E+04
90	0.647	-1.37E+08	-3.85E+08	1.37E+08	3.85E+08
100	0.648	9.58E+10	-2.28E+11	-9.58E+10	2.28E+11

Table 19 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=25$; Lethal Radius 5)

Lethal Radius	5				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.004	0.004	0.004	0.000	0.000
10	0.038	0.038	0.038	0.000	0.000
20	0.073	0.072	0.072	0.001	0.001
30	0.104	0.103	0.103	0.001	0.001
40	0.131	0.130	0.131	0.001	0.001
50	0.156	0.155	0.156	0.001	0.001
60	0.179	0.178	0.179	0.001	0.001
70	0.198	0.199	0.200	-0.001	-0.002
80	0.219	0.218	0.219	0.001	0.001
90	0.238	0.236	0.236	0.002	0.001
100	0.252	0.252	0.252	0.000	0.000

Table 20 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=25$; Lethal Radius 10)

Lethal Radius	10				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.016	0.016	0.016	0.000	0.000
10	0.131	0.131	0.132	0.001	0.000
20	0.220	0.220	0.222	0.000	-0.001
30	0.283	0.283	0.285	0.000	-0.002
40	0.331	0.331	0.333	0.001	-0.002
50	0.368	0.367	0.370	0.001	-0.002
60	0.391	0.396	0.399	-0.005	-0.008
70	0.418	0.420	0.423	-0.002	-0.005
80	0.437	0.440	0.443	-0.003	-0.006
90	0.453	0.457	0.460	-0.004	-0.007
100	0.468	0.472	0.475	-0.004	-0.007

Table 21 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=25$; Lethal Radius 20)

Lethal Radius	20				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.061	0.058	0.060	0.003	0.001
10	0.345	0.343	0.352	0.002	-0.008
20	0.459	0.463	0.475	-0.004	-0.016
30	0.515	0.526	0.539	-0.012	-0.025
40	0.555	0.567	0.580	-0.012	-0.025
50	0.581	0.596	0.610	-0.015	-0.029
60	0.601	0.619	0.632	-0.017	-0.031
70	0.620	0.637	0.650	-0.016	-0.030
80	0.635	0.652	0.665	-0.016	-0.030
90	0.646	0.664	0.678	-0.018	-0.032
100	0.656	0.675	0.689	-0.019	-0.033

Table 22 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=25$; Lethal Radius 30)

Lethal Radius	30				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.135	0.118	0.126	0.017	0.009
10	0.507	0.509	0.536	-0.001	-0.028
20	0.605	0.621	0.650	-0.016	-0.045
30	0.652	0.676	0.705	-0.024	-0.053
40	0.686	0.710	0.738	-0.024	-0.053
50	0.707	0.734	0.762	-0.027	-0.055
60	0.723	0.752	0.780	-0.029	-0.056
70	0.735	0.767	0.794	-0.031	-0.058
80	0.747	0.778	0.805	-0.031	-0.058
90	0.755	0.788	0.815	-0.033	-0.059
100	0.765	0.796	0.822	-0.032	-0.057

Table 23 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=25$; Lethal Radius 40)

Lethal Radius	40				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.226	0.184	0.204	0.041	0.022
10	0.630	0.633	0.681	-0.003	-0.051
20	0.715	0.734	0.781	-0.019	-0.065
30	0.756	0.781	0.825	-0.025	-0.069
40	0.781	0.809	0.851	-0.029	-0.070
50	0.797	0.829	0.868	-0.032	-0.071
60	0.810	0.844	0.881	-0.034	-0.071
70	0.819	0.855	0.891	-0.036	-0.072
80	0.828	0.865	0.899	-0.037	-0.071
90	0.835	0.877	0.861	-0.041	-0.026
100	0.842	0.933	-5.13E+00	-0.092	5.97E+00

Table 24 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=25$; Lethal Radius 50)

Lethal Radius	50				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.331	0.249	0.286	0.082	0.045
10	0.732	0.725	0.792	0.007	-0.060
20	0.801	0.815	0.872	-0.015	-0.072
30	0.832	0.854	0.905	-0.023	-0.073
40	0.849	0.877	0.923	-0.028	-0.074
50	0.864	0.893	0.935	-0.029	-0.071
60	0.872	0.904	0.943	-0.032	-0.071
70	0.881	0.909	0.943	-0.028	-0.063
80	0.886	0.913	0.565	-0.026	0.321
90	0.893	-2.80E+01	1.79E+01	2.89E+01	-1.70E+01
100	0.896	3.83E+03	1.28E+03	-3.83E+03	-1.28E+03

Table 25 Salvo Sim. vs. Approximation: ($\sigma_{\text{aiming}}=50$; $\sigma_{\text{bd}}=25$; Lethal Radius 60)

Lethal Radius	60				
#/Salvo	Sim Circle	Eq 16	Eq 17	Delta: Sim-Eq 16	Delta: Sim-Eq 17
1	0.438	0.306	0.365	0.132	0.072
10	0.812	0.793	0.871	0.019	-0.059
20	0.865	0.871	0.931	-0.006	-0.066
30	0.888	0.903	0.953	-0.016	-0.065
40	0.902	0.921	0.964	-0.019	-0.062
50	0.911	0.933	0.971	-0.022	-0.060
60	0.919	0.942	0.975	-0.023	-0.056
70	0.925	0.824	0.853	0.100	0.072
80	0.929	0.457	-4.68E+00	0.473	5.610
90	0.932	-3.96E+03	-4.90E+03	3.96E+03	4.90E+03
100	0.934	1.26E+05	-5.47E+05	-1.26E+05	5.47E+05

Two specific results can be observed from the tables above. First, it is clear that the approximations in Eq. (16) and Eq. (17) will break down at high values of number of bombs per salvo. This behavior is caused by the “battle of big alternating binomial coefficients” that results for high values of n [2]. The sudden breakdown of weapon effectiveness values calculated from Eq. (16) and Eq. (17) demonstrates one of the risks of using such approximations.

Second, the approximations performed rather poorly for Table 12 thru Table 18 when the ballistic dispersion was significantly smaller than aiming error. This demonstrates a more significant shortcoming of the approximations. This scenario is becoming more and more common as guided weapons become extremely accurate and self-designation by the fighter/bomber can cause a relatively large aiming error.

Finally, with the capability of modern computing systems and a significant level of uncertainty regarding the outputs of Eq. (16) and Eq. (17), it is recommended to use the separate error salvo weapon effectiveness procedure as detailed in Figure 11 whenever possible.

B. SQUARE TARGET

The data in the previous section used a circular MAE. It is possible that a weapon effective area is more rectangular in nature. More of the weapon effectiveness is focused out of the side of the munition casing. If a bomb impacts with a shallow angle the area of effectiveness is longer in the deflection direction and shorter in the range direction. In the absence of a salvo weapon effectiveness equation/approximation the previous weapon effectiveness procedure from Figure 11 can be altered to check if the weapon impacts a rectangular area centered on the target. This again demonstrates the versatility of using the simulation procedure to calculate weapon effectiveness parameters.

The following results show the weapon effectiveness values for a circle of radius 25 compared to various aspect ratio rectangles of the same area. Length to width ratios of 1, 0.5, 0.25, 0.125, and 0.0625 will be evaluated for the $\sigma_{aiming}=50$; $\sigma_{bd}=25$; Lethal Radius 20 scenario.

Table 26 Circular vs. Rectangle Weapon Effectiveness Area Salvo Simulation

$\sigma_{aiming}=50$; $\sigma_{bd}=25$; Lethal Radius 20						
	Ratio:	1	0.5	0.25	0.125	0.0625
#/Salvo	Circle	Square	Rectangle	Rectangle	Rectangle	Rectangle
1	0.062	0.062	0.062	0.059	0.056	0.050
10	0.346	0.509	0.504	0.488	0.452	0.397
20	0.458	0.616	0.617	0.610	0.596	0.553
30	0.517	0.667	0.670	0.669	0.660	0.628
40	0.554	0.700	0.701	0.704	0.694	0.671
50	0.581	0.722	0.723	0.725	0.718	0.701
60	0.602	0.739	0.743	0.743	0.737	0.722
70	0.621	0.755	0.755	0.755	0.752	0.737
80	0.633	0.763	0.765	0.769	0.765	0.751
90	0.646	0.774	0.775	0.778	0.775	0.760
100	0.658	0.782	0.782	0.784	0.782	0.771

The weapon effectiveness comparisons shown in Table 26 detail a scenario for a circular distribution of impacts with a rectangular MAE for the weapon. The rectangular MAE results in slightly higher weapon effectiveness than the circular MAE of equivalent area. Also of interest are the slightly smaller effectiveness parameters that result from the reduced MAE aspect. This implies, for this scenario, as the rectangular MEA becomes thinner and longer the weapon effectiveness is slightly reduced. With minor modifications these techniques can also be used to calculate weapon effectiveness for non-circular impact distributions with any weapon/target MAE shape desired.

V. ATTEMPTS TO EXTRACT ERROR TYPES FROM IMPACT DATA

Often during weapon delivery performance testing the primary data gathered are the range and deflection miss distances. Having the weapon impact miss distances and an estimation of the type of expected distribution one would hope to be able to extract the aiming error and ballistic dispersion error values that created the impact locations. Unfortunately, for a normal distribution this extraction is ambiguous due to the RSS value of the aiming and ballistic dispersion errors allowing for an infinite number of combinations for the same distribution. However, there is still value added in attempting to properly characterize the weapon impact distribution for the purpose of weapon effectiveness calculations. Complicating this process is the fact that guided weapon distributions are typically not normal distributions. Therefore, a method to approximate an appropriate distribution for a non-normal dataset must be introduced.

A. ALGORITHM DESCRIPTION

This process starts by assuming a particular form for the weapon miss distance CDF. For the purposes of this discussion, it will be assumed that the non-normal distribution is an expression derived from linear combinations of other known distributions weighted accordingly. These distributions will be discussed in more detail in Section B. Once a distribution is defined, the impact data needs to be rank ordered and a CDF needs to be calculated from the impact data. Once this empirically derived CDF is determined a least-squares-fit to the assumed CDF is performed using MATLAB. The parameters being used to accomplish the curve fit are the standard deviations of the CDF and the weighting factors. Once the weighting factors and standard deviations are known, the weapon effectiveness calculations can be performed. See Figure 12 for a diagram of this process. See Section B for examples. [4]

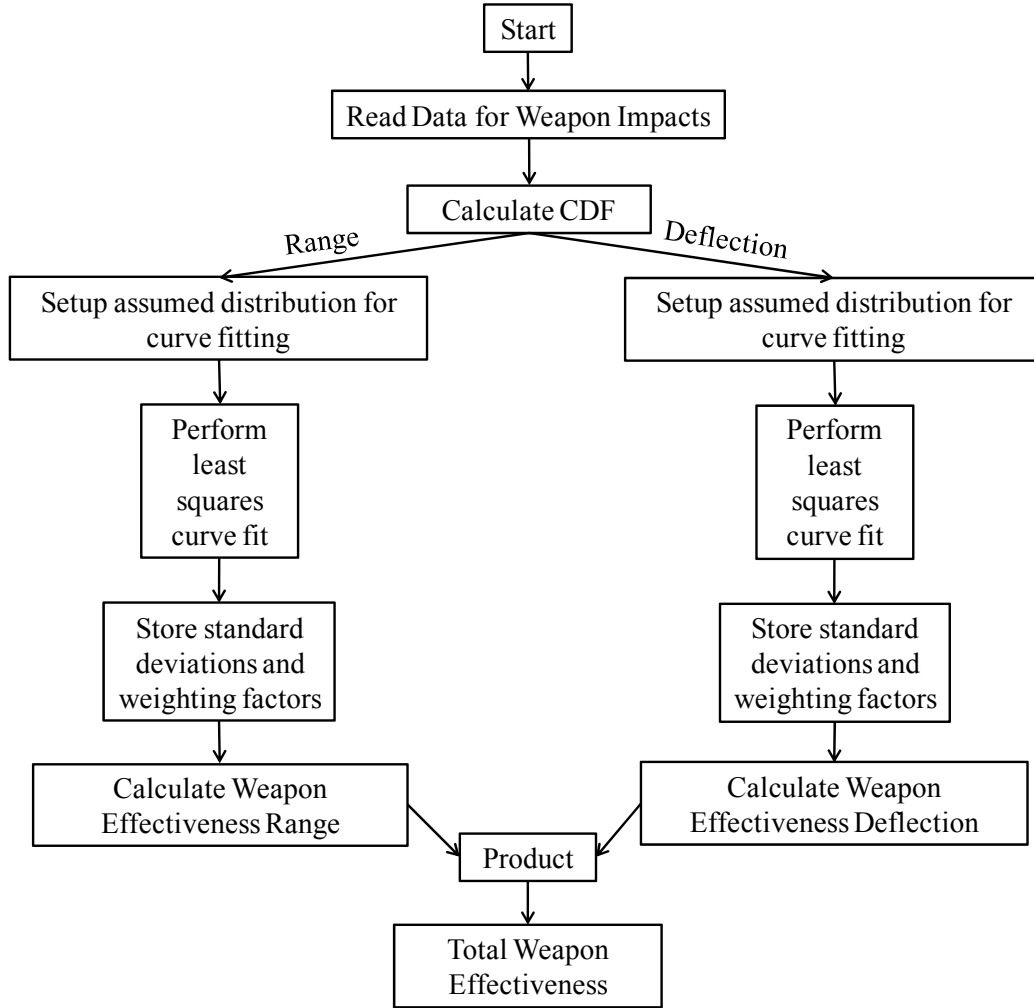


Figure 12 Flow Chart for Extracting Weapon Effectiveness From Impact Data

B. SINGLE ROUND SCENARIO

1. Double Normal Approximation to Non-Normal Dataset

For an assumed double normal dataset the distribution would be defined as follows. Knowing that the univariate normal PDF is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} * e^{\left[\frac{-(x-\mu)^2}{2\sigma^2}\right]} \quad (22)$$

And the univariate normal CDF is:

$$F(x) = \int_{-\infty}^x \frac{1}{\sigma\sqrt{2\pi}} * e^{\left[\frac{-(x-\mu)^2}{2\sigma^2}\right]} dx = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{z}{\sqrt{2}} \right) \right] \quad (23)$$

$$\text{Where: } z = \frac{x - \mu}{\sigma}$$

Creating the double normal CDF in the range direction yields:

$$\begin{aligned} CDF_{DN_x} &= w_{x1} * \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{z_1}{\sqrt{2}} \right) \right] + w_{x2} * \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{z_2}{\sqrt{2}} \right) \right] \\ CDF_{DN_x} &= w_{x1} * \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x}{\sigma_{x1}\sqrt{2}} \right) \right] + w_{x2} * \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x}{\sigma_{x2}\sqrt{2}} \right) \right] \end{aligned} \quad (24)$$

$$\text{Where: } w_{x1} + w_{x2} = 1$$

Substituting $w_{x2} = 1 - w_{x1}$ yields:

$$CDF_{DN_x} = w_{x1} * \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x}{\sigma_{x1}\sqrt{2}} \right) \right] + (1 - w_{x1}) * \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x}{\sigma_{x2}\sqrt{2}} \right) \right] \quad (25)$$

This is the function to be used by the curve fitting routine to extract $w_{x1}, \sigma_{x1}, \sigma_{x2}$.

Similarly for the deflection direction:

$$CDF_{DN_y} = w_{y1} * \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{y}{\sigma_{y1}\sqrt{2}} \right) \right] + (1 - w_{y1}) * \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{y}{\sigma_{y2}\sqrt{2}} \right) \right] \quad (26)$$

This is the function to be used by the curve fitting routine to extract $w_{y1}, \sigma_{y1}, \sigma_{y2}$.

As a test of this algorithm, a dataset will be created using Eq. (25). Setting $w_{x1} = 0.7$, $\sigma_{x1} = 30$, $\sigma_{x2} = 5$ over the range $[-100:2:100]$ and adding a small error source to randomize the dataset results in the dataset shown in Figure 13.

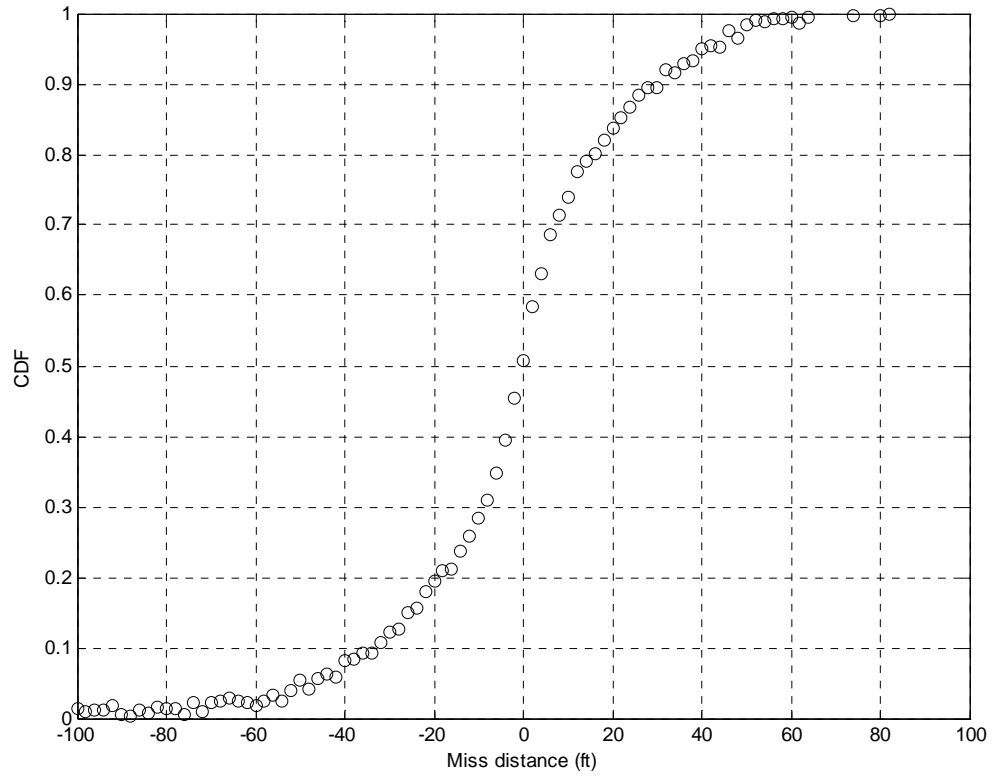


Figure 13 Artificially Created Double Normal CDF ($w_{x1} = 0.3, \sigma_{x1} = 30, \sigma_{x2} = 5$)

The next step is to curve fit the data to extract $w_{x1}, \sigma_{x1}, \sigma_{x2}$. Using initial conditions of:

$$\begin{array}{ll} w_{x1} = 0.1 & w_{x1} = 0.317 \\ \sigma_{x1} = 5 & \text{results in curve fitting estimates of: } \sigma_{x1} = 5.693 \\ \sigma_{x2} = 40 & \sigma_{x2} = 30.407 \end{array}$$

The curve created by these estimates can be seen in comparison with the test dataset on Figure 14.

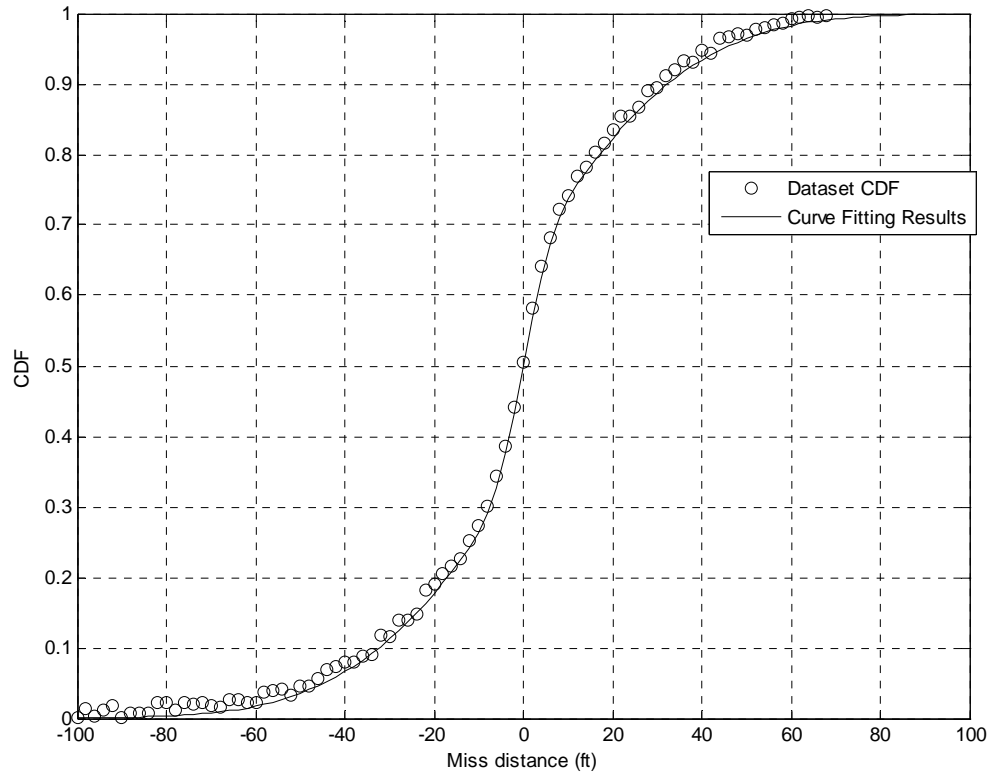


Figure 14 Comparison of CDFs: Double Normal Dataset vs. Curve Fitting

It is clear for this test case the algorithm curve fits the dataset reasonably well. It is interesting to note that σ_{x1} and σ_{x2} are flipped due to the procedure converging to a w_{x1} of 0.317 instead of the actual value of 0.7. This potential swapping of values between the two standard deviations will not have an effect of future weapon effectiveness calculations as the weights associated with them are also swapped. A similar process can be performed for the deflection direction.

a. Double Normal Dataset Weapon Effectiveness Calculations

Once σ_{x1} , σ_{x2} , σ_{y1} , and σ_{y2} have been extracted from the non-normal datasets in the range and deflection direction the weapon effectiveness calculations can be performed. Again, a MAE will be required to perform the weapon effectiveness simulations. This MAE will be assumed to be rectangular with length (in the range direction) L_{ET} and width (in the deflection direction) W_{ET} .

The range direction calculations for weapon effectiveness can again be performed using Monte-Carlo simulations. Each range standard deviation (σ_{x1} and σ_{x2}) can be simulated individually to determine the weapon effectiveness resulting from each parameter. These Monte-Carlo simulations will be performed as outlined on the right side of Figure 11. To determine if an individual weapon is a kill the impact location is compared to $L_{ET}/2$. If inside $L_{ET}/2$ the weapon is considered a kill. These two simulations result what is called a single sortie probability of damage (SSPD). In the range direction the two SSPDs will be combined using Eq. (27) to yield a total SSPD in the range direction.

$$SSPD_{x_{Total}} = w_{x1} * SSPD_{x1} + (1 - w_{x1}) * SSPD_{x2} \quad (27)$$

Following a similar procedure for the deflection direction using $W_{ET}/2$ to determine a kill results in Eq. (28).

$$SSPD_{y_{Total}} = w_{y1} * SSPD_{y1} + (1 - w_{y1}) * SSPD_{y2} \quad (28)$$

To complete the weapon effectiveness calculation the total SSPD is defined as Eq. (29).

$$SSPD_{TOTAL} = SSPD_{x_{TOTAL}} * SSPD_{y_{TOTAL}} \quad (29)$$

As an example, using the previous curve fitted values of $w_{x1} = 0.317$, $\sigma_{x1} = 5.693$, and $\sigma_{x2} = 30.407$ and running the Monte Carlo simulations for the range direction assuming $L_{ET}=20$ and $W_{ET}=20$ results in:

$$\begin{aligned}
SSPD_{x1} &= 0.922 \\
SSPD_{x2} &= 0.257 \\
SSPD_{x_{Total}} &= w_{x1} * SSPD_{x1} + (1 - w_{x1}) * SSPD_{x2} \\
SSPD_{x_{Total}} &= 0.317 * 0.922 + (1 - 0.317) * 0.257 = 0.47
\end{aligned} \tag{30}$$

Assuming the distribution is circular results in:

$$\begin{aligned}
SSPD_{TOTAL} &= SSPD_{x_{TOTAL}} * SSPD_{y_{TOTAL}} = SSPD_{x_{TOTAL}}^2 \\
SSPD_{TOTAL} &= 0.47^2 = 0.22
\end{aligned} \tag{31}$$

2. Double Rayleigh Approximation to Non-Normal Dataset

The procedure is similar to the one describe for a double normal distribution but instead a linear combination of Rayleigh distributions is used. Knowing the Rayleigh PDF is:

$$f(r) = \frac{r}{\sigma^2} e^{\left[\frac{-r^2}{2\sigma^2}\right]} \tag{32}$$

And the Rayleigh CDF is:

$$F(R) = 1 - e^{\left[\frac{-R^2}{2\sigma^2}\right]} \tag{33}$$

Substituting this expression into the correct terms of Eq. (25) yields the double Rayleigh distribution in Eq. (34).

$$CDF_{DR} = w_1 * \left[1 - e^{\left(\frac{-R^2}{2\sigma_1^2}\right)} \right] + (1 - w_1) * \left[1 - e^{\left(\frac{-R^2}{2\sigma_2^2}\right)} \right] \tag{34}$$

As a test of this algorithm, a dataset will be created using Eq. (34). Setting $w_{x1} = 0.7$, $\sigma_{x1} = 30$, $\sigma_{x2} = 5$ over the range $[0:2:100]$ and adding a small error source to randomize the dataset results in the dataset shown in Figure 15.

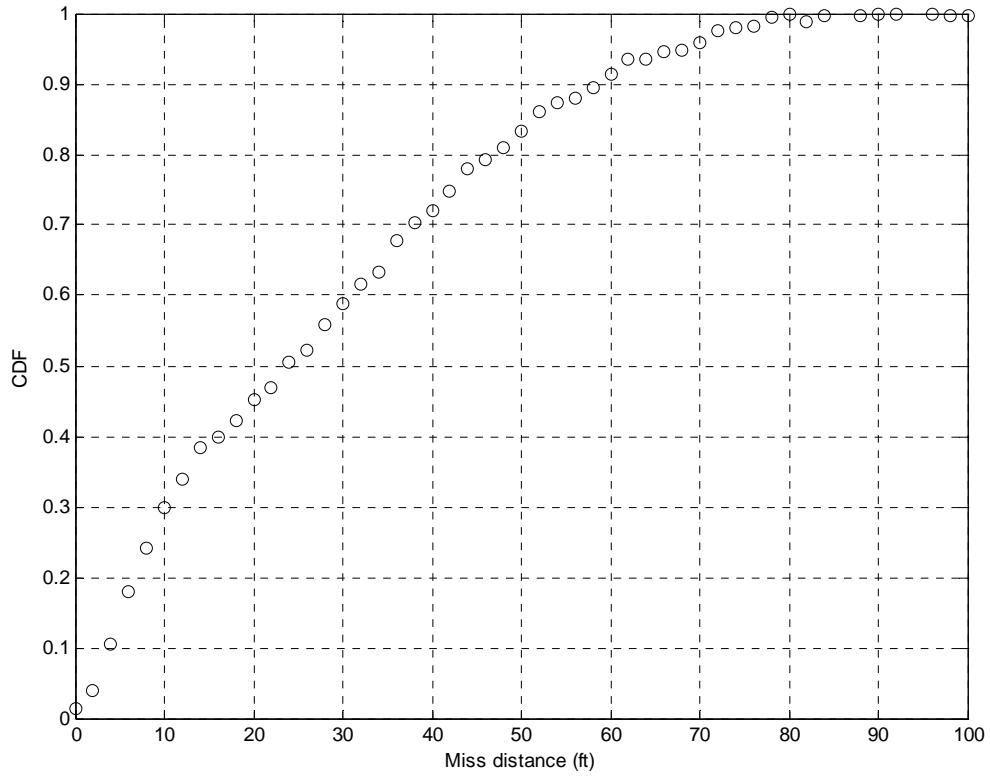


Figure 15 Artificially Created Double Rayleigh CDF ($w_{r1} = 0.3, \sigma_{r1} = 30, \sigma_{r2} = 5$)

The next step is to curve fit the data to extract $w_{r1}, \sigma_{r1}, \sigma_{r2}$. Using initial conditions of:

$$\begin{array}{ll}
 w_{r1} = 0.1 & w_{r1} = 0.300 \\
 \sigma_{r1} = 5 & \text{results in curve fitting estimates of: } \sigma_{r1} = 4.794 \\
 \sigma_{r2} = 40 & \sigma_{r2} = 29.157
 \end{array}$$

The curve created by these estimates can be seen in comparison with the test dataset on Figure 16.

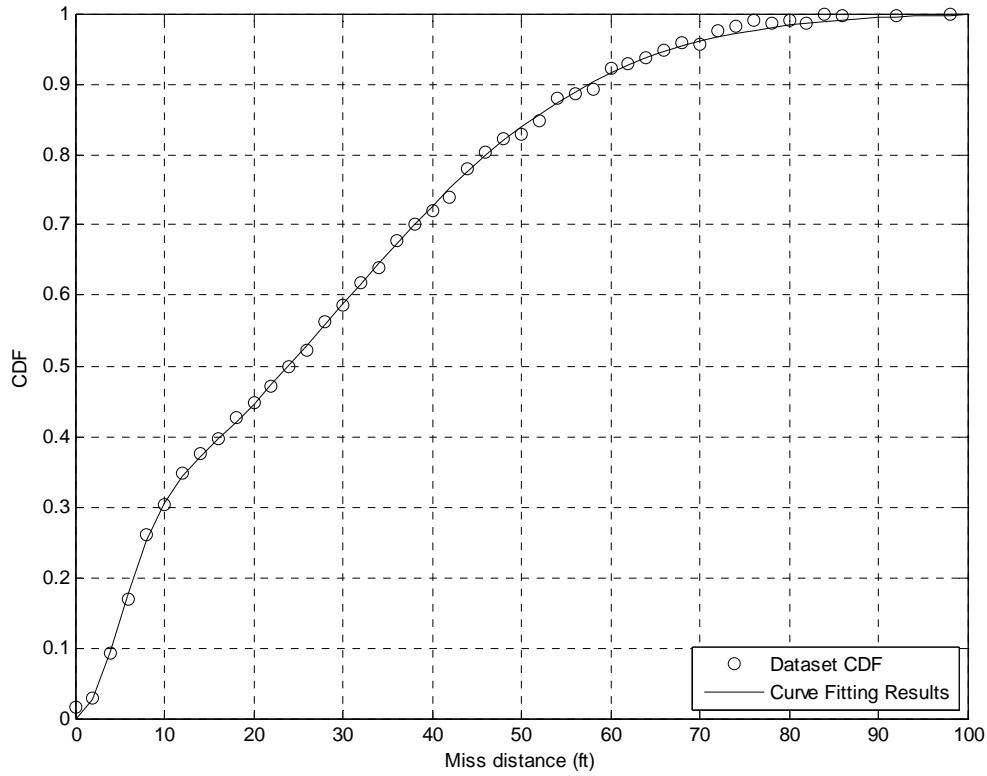


Figure 16 Comparison of CDFs: Double Rayleigh Dataset vs. Curve Fitting

Again, similarly to the double normal distribution extraction tests, the double Rayleigh procedure produced a reasonable curve fit and extracted values for the standard deviation and weighting value close to the original values. These values can now be used to perform weapon effectiveness calculations.

a. Double Rayleigh Dataset Weapon Effectiveness Calculations

The weapon effectiveness calculations for the double Rayleigh distribution are more straightforward than those for the double normal distribution because there is only one axis to analyze. Therefore, a simulation is made for each standard deviation to generate a radial miss distance and compare this with a predefined lethal radius. Each of these simulations will output an $SSPD_r$. Once both $SSPD_{r1}$ and $SSPD_{r2}$ are simulated the total SSPD can be calculated as shown in Eq. (35).

$$SSPD_{r_{Total}} = w_{r1} * SSPD_{r1} + (1 - w_{r1}) * SSPD_{r2} \quad (35)$$

As an example, using the previous curve fitted values of $w_{x1} = 0.300$, $\sigma_{x1} = 4.794$, and $\sigma_{x2} = 29.157$ and running the Monte Carlo simulations for the radial direction assuming Weapon Lethal Radius=11.28 results in:

$$\begin{aligned} SSPD_{r1} &= 0.937 \\ SSPD_{r2} &= 0.073 \\ SSPD_{TOTAL} &= w_{r1} * SSPD_{r1} + (1 - w_{r1}) * SSPD_{r2} \\ SSPD_{TOTAL} &= 0.300 * 0.937 + (1 - 0.300) * 0.073 = 0.33 \end{aligned} \quad (36)$$

VI. CONCLUSIONS AND RECOMMENDATIONS

From the numerical investigations presented above, the following can be concluded:

1. Extraction of aiming error and ballistic dispersion error from single weapon impact test data is not possible due to an infinite number of combinations of errors that will yield the same distribution.
2. If salvo weapon impact data is available, aiming error and ballistic dispersion error extraction would be possible.
3. Aiming error and ballistic dispersion error can be combined using a root-sum-square for the purposes of accuracy parameter calculations and simulations.
4. The aiming error and ballistic dispersion cannot be combined using a root-sum-square for the purposes of weapon effectiveness calculations.
5. Monte-Carlo simulation for salvo weapon effectiveness can provide more accurate results than salvo equation approximations
6. For single round scenarios it is possible to extract two standard deviations with associated weighting values and calculate weapon effectiveness for a single round.

The following are recommendations for future research:

1. For salvo weapon impact data error extraction, investigate the required number of bombs per salvo and number of salvos required to yield accurate aiming error and ballistic dispersion error.
2. Investigate the use of MATLAB Symbolic Toolbox to provide numerical answers to derived analytical expressions for weapon effectiveness.

3. Investigate the potential of MATLAB Statistics Toolbox to identify dataset characteristics to create more accurate non-normal distribution functions to be used for standard deviation extraction.
4. Continue researching the sensitivity of the salvo weapon effectiveness result by varying the ratio of aiming error to ballistic dispersion error, the number of weapons per salvo, and the lethal radius.

APPENDIX. MATLAB CODE

A. CHAPTER II CODE

1. Code for Table 2

```
clear
sigmax=50

%REP calculation
bombnumber=1000000;
x=ones(bombnumber,10);
for k=1:1:10
    i=1;
    while i<=bombnumber
        x(i,k)=randn*sigmax;
        i=i+1;
    end
end
REP=median(abs(x))
```

2. Code for Table 3

```
clear
sigmax=50;
sigmay=50;

%REP calculation
bombnumber=10000000;
x=ones(bombnumber,1);
y=ones(bombnumber,1);
r=ones(bombnumber,1);
for k=1:1:1
    i=1;
    while i<=bombnumber
        x(i)=randn*sigmax;
        y(i)=randn*sigmay;
        r(i)=sqrt(x(i)^2+y(i)^2);
        i=i+1;
    end
end
CEP=median(abs(r))
```

3. Code for Table 4

```
% Code to calculate the Pk with aiming error (oto),
% ballistic dispersion (rtr), warhead radius
```

```

clear
%Ocasión-to-ocasión sigma
sigmaoto=30;
%Round-to-round sigma
sigmartr=10;
%Warhead Leathal Range
warheadLR=40;
%Num of Bombs Per Occasion
numbpoc=5;
occ=10;
kVector=ones(occ,1);

for i=1:occ %seperate occasions to be simulated
    %x1 and y1 are the OCO errors
    x1=randn*sigmaoto;
    y1=randn*sigmaoto;
    j=1;
    while j<=numbpoc
        %x2 and y2 are the RTR errors
        x2=randn*sigmartr;
        y2=randn*sigmartr;
        %xt and yt are total error vectors (one col. per bomb)
        xt(j)=x1+x2;
        yt(j)=y1+y2;
        rMiss(j)=(xt(j)^2+yt(j)^2)^.5;
        j=j+1;
    end

    %If rMiss is inside tgt dimensions
    %then a hit has occurred
    if min(rMiss)<=warheadLR
        kVector(i)=1;
    else
        kVector(i)=0;
    end
end
%Calc Pk from kVector
Pk=sum(kVector)/length(kVector)

```

B. CHAPTER III CODE

1. Code for Table 5-Table 9

```

% Investigation into similarity of OTO and RTR as split sigmas vs. one
% total sigma
clear
%Ocasión-to-ocasión sigma
sigmaoto=210
%Round-to-round sigma
sigmartr=100
numoc=1000000;
%Number of bombs per occasions
numbpoc=7;

```

```

for q=1:1:10
    %Split sigmas calculation
    clear x1 x2 x3 y1 y2 y3 xt yt r1 r2
    xt=ones(numoc*numbpoc,1);
    yt=ones(numoc*numbpoc,1);
    r2=ones(numoc*numbpoc,1);
    x3=ones(numoc*numbpoc,1);
    y3=ones(numoc*numbpoc,1);
    r1=ones(numoc*numbpoc,1);

    bombnumber=0;
    i=1;
    while i<=numoc
        x1=randn*sigmaoto;
        y1=randn*sigmaoto;
        j=1;
        while j<=numbpoc
            bombnumber=bombnumber+1;
            x2=randn*sigmartr;
            y2=randn*sigmartr;
            xt(bombnumber)=x1+x2;
            yt(bombnumber)=y1+y2;
            r2(bombnumber)=(xt(bombnumber)^2+yt(bombnumber)^2)^.5;
            j=j+1;
        end
        i=i+1;
    end
    cepTwoSigmas(q)=median(r2);
    DEPTwoSigmas(q)=median(abs(xt));
    REPTwoSigmas(q)=median(abs(yt));

    %Total sigma calculation
    sigmat=(sigmaoto^2+sigmartr^2)^.5;
    k=1;
    while k<=bombnumber
        x3(k)=randn*sigmat;
        y3(k)=randn*sigmat;
        r1(k)=(x3(k)^2+y3(k)^2)^.5;
        k=k+1;
    end
    cepOneSigma(q)=median(r1);
    DEPOneSigma(q)=median(abs(x3));
    REPOneSigma(q)=median(abs(y3));
end

cep2avg=mean(cepTwoSigmas)
ceplavg=mean(cepOneSigma)
deltaCep=ceplavg-cep2avg
CEPeql4=1.1774*sigmat

REP2avg=mean(REPTwoSigmas)
REPlavg=mean(REPOneSigma)
deltaREP=REPlavg-REP2avg

```

```

REPeql2=0.6745*sigmat

DEP2avg=mean(DEPTwoSigmas)
DEP1avg=mean(DEPOneSigma)
deltaDEP=DEP1avg-DEP2avg
DEPeql3=0.6745*sigmat

percenterrorCEP=deltaCep/cep2avg
percenterrorREP=deltaREP/REP2avg
percenterrorDEP=deltaDEP/DEP2avg

```

2. Code for Table 10-Table 11

```

% Code to calculate the Pk with aiming error (oto),
% ballistic dispersion (rtr), warhead radius

clear
%Occasion-to-occasion sigma
sigmaoto=30;
%Round-to-round sigma
sigmartr=10;
%Warhead Leathal Range
warheadLR=40;
%Num of Bombs Per Occasion
numbpoc=5;
occ=10;
kVector=ones(occ,1);

for i=1:occ %seperate occasions to be simulated
    %x1 and y1 are the OCO errors
    x1=randn*sigmaoto;
    y1=randn*sigmaoto;
    j=1;
    while j<=numbpoc
        %x2 and y2 are the RTR errors
        x2=randn*sigmartr;
        y2=randn*sigmartr;
        %xt and yt are total error vectors (one col. per bomb)
        xt(j)=x1+x2;
        yt(j)=y1+y2;
        rMiss(j)=(xt(j)^2+yt(j)^2)^.5;
        j=j+1;
    end

    %If rMiss is inside tgt dimensions
    %then a hit has occured
    if min(rMiss)<=warheadLR
        kVector(i)=1;
    else
        kVector(i)=0;
    end
end

```

```

        %Calc Pk from kVector
        Pk=sum(kVector)/length(kVector)

%THIS CODE IS THE INCORRECT METHOD FOR WEAPON EFFECTIVENESS%
clear
    %Ocasion-to-ocasion sigma
    sigmaoto=50
    %Round-to-round sigma
    sigmartr=5
    %Warhead Leathal Range
    warheadLR=50
    %number of occasions
    numoc=100000
    %RSS of errors
    sigmat=sqrt(sigmaoto^2+sigmartr^2)
    %desired pk
    pkdesired=0.7
    %number of rounds to power up (this is the incorrect method!!!)
    ndesired=10
for q=1:10
rMiss=zeros(numoc,1);
    for i=1:numoc%seperate weapons to be simulated
        %x1 and y1 are the OCO errors
        xt=randn*sigmat;
        yt=randn*sigmat;
        rMiss(i)=(xt^2+yt^2)^.5;
        i=i+1;
    end

    %Calc Pk from kVector
    Pk(q)=sum(rMiss<=warheadLR)/length(rMiss);
end
Pkavg=mean(Pk)

%now calc required number of bombs to achieve pk=0.7 solve power-up
formula
n=log(1-pkdesired)/log(1-Pkavg)

PkpoweredUP=1-(1-Pkavg)^ndesired
%THIS CODE IS THE INCORRECT METHOD FOR WEAPON EFFECTIVENESS%

```

C. CHAPTER IV CODE

1. Code for Table 12-Table 25

```

% Code to calculate the Pk with oto, rtr, tgtDim (square and circle)

clear
    %Ocasion-to-ocasion sigma
    sigmaoto=164.469
    %Round-to-round sigma
    sigmartr=164.469

```

```

%Target Square Dimension Conversion from warhead lethal raduis
warheadLR=150
tgtX=warheadLR*pi^.5
tgtY=warheadLR*pi^.5
    %Calculations for half tgt side length used to determine a hit
    tgthalfX=tgtX/2
    tgthalfY=tgtY/2
    %One Salvo or number of occasions
    numoc=200000
for numbpoc=7
    for q=1:10
        clear xt yt kVectorsquare kVectorcircle rMiss
        xt=zeros(numbpoc,1);
        yt=zeros(numbpoc,1);
        rMiss=zeros(numbpoc,1);
        kVectorsquare=zeros(numoc,1);
        kVectorcircle=zeros(numoc,1);
            for i=1:numoc %seperate occasions to be simulated
                %x1 and y1 are the OCO errors
                x1=randn*sigmaoto;
                y1=randn*sigmaoto;
                j=1;
                while j<=numbpoc
                    %x2 and y2 are the RTR errors
                    x2=randn*sigmartr;
                    y2=randn*sigmartr;
                    %xt and yt are total error vectors (one col. per bomb)
                    xt(j)=x1+x2;
                    yt(j)=y1+y2;
                    rMiss(j)=(xt(j)^2+yt(j)^2)^.5;
                    j=j+1;
                end
                %If absolute value xt and yt are inside tgt dimensions
                %then a hit has occurred

kVectorsquare(i)=and(min(abs(xt))<=tgthalfX,min(abs(yt))<=tgthalfY);
        kVectorcircle(i)=min(rMiss)<=warheadLR;
        end
        %Calc Pk from kVector
        Pksquare=sum(kVectorsquare)/length(kVectorsquare);
        Pkcircle=sum(kVectorcircle)/length(kVectorcircle);
        %store results from the above
        PkVectorsquare(q)=Pksquare;
        PkVectorcircle(q)=Pkcircle;
    end
    %average a number of runs to calculate final expected Pk
    Pkavgsquare(numbpoc,1)=mean(PkVectorsquare);
    Pkavgcircle(numbpoc,1)=mean(PkVectorcircle);
end
Pkavgsquare
Pkavgcircle

clear
c=1 % correction factor between .9 and 1 ???

```

```

Rt=200
sigmax=50 %rtr dispersion
sigmau=150 %oco aiming error
P=zeros(20,1);
P2=zeros(20,1);
for n=1:20
%eqn 20-11
    for i=1:n
        Pn=(-1)^(i+1)*factorial(n)/(factorial(i)*factorial(n-
i))*(1/i)*((Rt^2/sigmax^2)/(2*c+Rt^2/sigmax^2))^(i-1)*(1-exp(-
1*(Rt^2/sigmax^2)/((2*c+Rt^2/sigmax^2)/(c*i)+2*sigmau^2/sigmax^2)));
        P(n)=P(n)+Pn;
    end

%eqn 20-13
    for i=1:n
        Pn2=(-1)^(i+1)*factorial(n)/(factorial(i)*factorial(n-
i))*(Rt^2/(sigmax^2*(2+Rt^2/sigmax^2)))^(i-
1)*((Rt^2/sigmax^2)/(2+Rt^2/sigmax^2+i*2*sigmau^2/sigmax^2));
        P2(n)=P2(n)+Pn2;
    end
end
P
P2

```

2. Code for Table 26

```

% Code to calculate the Pk with oto, rtr, tgtDim (square and circle)

clear
%Occasion-to-occasion sigma
sigmaoto=210
%Round-to-round sigma
sigmartr=100
%calculate RSS sigma
sigmarss=(sigmaoto^2+sigmartr^2)^.5
%Target Square Dimension Conversion from warhead lethal raduis
warheadLR=220
tgtX=warheadLR*pi^.5
tgtY=warheadLR*pi^.5
    %Calculations for half tgt side length used to determine a hit
    tgthalfX=tgtX/2
    tgthalfY=tgtY/2
%One Salvo or number of occasions
for numbpoc=1:10
    for q=1:10
        clear x1 x2 y1 y2 xt yt kVectorsquare kVectorcircle rMiss
            for i=1:20000 %seperate occasions to be simulated
                j=1;
                while j<=numbpoc
                    %xt and yt are total error vectors (one col. per bomb)
                    xt(j)=randn*sigmarss;
                    yt(j)=randn*sigmarss;
                    rMiss(j)=(xt(j)^2+yt(j)^2)^.5;
                    j=j+1;
                end
            end
        end
    end

```

```

        end
        %If absolute value xt and yt are inside tgt dimensions
        %then a hit has occurred

kVectorsquare(i)=and(min(abs(xt))<=tgthalfX,min(abs(yt))<=tgthalfY);
kVectorcircle(i)=min(rMiss)<=warheadLR;
    end
    %Calc Pk from kVector
    Pksquare=sum(kVectorsquare)/length(kVectorsquare);
    Pkcircle=sum(kVectorcircle)/length(kVectorcircle);
    %store results from the above
    PkVectorsquare(q)=Pksquare;
    PkVectorcircle(q)=Pkcircle;
end
%average a number of runs to calculate final expected Pk
Pkavgsquare(numbpoc,1)=mean(PkVectorsquare);
Pkavgcircle(numbpoc,1)=mean(PkVectorcircle);
end
Pkavgsquare
Pkavgcircle

```

D. CHAPTER V CODE

1. Double Normal Approximation

```

% This program generates DN distributed data with noise and then fits a
double normal
% distribution in order to recover the original distribution parameters

%---curve fitting test program-----
clear
% First create the data.
t=-100:2:100;
t=t(:); % To make t a column vector
w1=.7
s1=30;s2=5;
Data=1/2*(w1*(1+erf(t/(s1*2^.5)))+(1-
w1)*(1+erf(t/(s2*2^.5))))+0.02*rand(size(t));
figure(1)
plot(t,Data,'o'); %plot data to be fitted
ylim([0,1])
hold on

% Now call FMINSEARCH.
Starting=[0.1 5 40];
options=optimset('Display','iter');
Estimates=fminsearch(@DN_myfit,Starting,options,t,Data);

% To check the fit
wle1=Estimates(1)
sle1=Estimates(2)
s2e1=Estimates(3)

```

```

Data_pred1=1/2*(wle1*(1+erf(t/(sle1*2^.5)))+(1-
wle1)*(1+erf(t/(s2e1*2^.5))));% function fitted to data
plot(t,Data_pred1,'r');
xlabel('Miss distance (ft)')
ylabel('CDF')
grid on
legend('Dataset CDF','Curve Fitting Results','Location','SouthEast')
hold off
% Weapon effectiveness calculation using extracted values and weight

Wet=20
Let=20

numbpoc=1
for q=1:10
clear x1 kVector
    for i=1:20000 %seperate occasions to be simulated
        %x1 and y1 are the OCO errors
        x1=randn*sle1;

        %If absolute value x1 is inside Wet dimension
        %then a hit has occured
        if abs(x1)<=Let/2
            kVector(i)=1;
        else
            kVector(i)=0;
        end
    end
    %Calc Pk from kVector
    Pk1=sum(kVector)/length(kVector);
    %store results from the above
    PkVector1(q)=Pk1;
end
%average a number of runs to calculate final expected Pk
SSPDx1=mean(PkVector1);
SSPDx1w=SSPDx1*wle1;

for q=1:10
clear x2 kVector
    for i=1:20000 %seperate occasions to be simulated
        %x1 and y1 are the OCO errors
        x2=randn*s2e1;

        %If absolute value x1 is inside Wet dimension
        %then a hit has occured
        if abs(x2)<=Let/2
            kVector(i)=1;
        else
            kVector(i)=0;
        end
    end
    %Calc Pk from kVector
    Pk2=sum(kVector)/length(kVector);
    %store results from the above

```

```

        PkVector2(q)=Pk2;
    end
    %average a number of runs to calculate final expected Pk
    SSPDx2=mean(PkVector2);
    SSPDx2w=SSPDx2*(1-w1e1);

    SSPDxTotal=SSPDx1w+SSPDx2w

    %Setup Parameters
    mu=0;
    s=s1e1;
    a=-Wet/2;
    b=Wet/2;

    %Calculate the probability a sample lies within the given range [a:b]
    syms x;
    normpdf=1/(s*sqrt(2*pi))*exp(-(x-mu).^2)/(2*s^2));
    Prob=int(normpdf,a,b);
    SSPD1analytical=double(Prob);
    SSPD1analyticalw=SSPD1analytical*w1e1;

    mu=0;
    s=s2e1;
    a=-Wet/2;
    b=Wet/2;

    %Calculate the probability a sample lies within the given range [a:b]
    syms x;
    normpdf=1/(s*sqrt(2*pi))*exp(-(x-mu).^2)/(2*s^2));
    Prob=int(normpdf,a,b);
    SSPD2analytical=double(Prob);
    SSPD2analyticalw=SSPD2analytical*(1-w1e1);
    SSPDxtotalanalytical=SSPD1analyticalw+SSPD2analyticalw

function sse=myfit(params,Input,Actual_Output)
wlf=params(1);
slf=params(2);
s2f=params(3);
Fitted_Curve=1/2*(wlf*(1+erf(Input/(slf*2^.5)))+(1-
wlf)*(1+erf(Input/(s2f*2^.5))));
Error_Vector=Fitted_Curve - Actual_Output;
% When curvefitting, a typical quantity to
% minimize is the sum of squares error
sse=sum(Error_Vector.^2);
% You could also write sse as
% sse=Error_Vector(:)'*Error_Vector(:);

```

2. Double Rayleigh Approximation

```
% This program generates DR distributed data with noise and then fits a
double normal
% distribution in order to recover the original distribution parameters

%---curve fitting test program-----
clear
% First create the data.
t=0:2:100;
t=t(:); % To make t a column vector
w1=.7
s1=30;s2=5;
Data=1-w1*exp(-t.*(2*s1*s1))-(1-w1)*exp(-
t.*(2*s2*s2))+0.02*rand(size(t));
figure(1)
plot(t,Data,'o'); %plot data to be fitted
ylim([0,1])
hold on

% Now call FMINSEARCH.
Starting=[0.1 5 40];
options=optimset('Display','iter');
Estimates=fminsearch(@DR_myfit,Starting,options,t,Data);

% To check the fit
wle1=Estimates(1)
sle1=Estimates(2)
s2e1=Estimates(3)
Data_pred1=1-wle1*exp(-t.*(2*sle1*sle1))-(1-wle1)*exp(-
t.*(2*s2e1*s2e1));% function fitted to data
plot(t,Data_pred1,'r');
xlabel('Miss distance (ft)')
ylabel('CDF')
grid on
legend('Dataset CDF','Curve Fitting Results','Location','SouthEast')
hold off

% Weapon effectiveness calculation using extracted values and weight

Wet=20
Let=20
LethalRadius=((Wet*Let)/pi)^0.5

numbpoc=1
for q=1:10
clear x1 kVector
for i=1:20000 %seperate occasions to be simulated
    %x1 and y1 are the OCO errors
    x1=randn*sle1;
    y1=randn*sle1;
    r1=sqrt(x1^2+y1^2);
    %If absolute value x1 is inside Wet dimension
    %then a hit has occured
```

```

        if r1<=LethalRadius
            kVector(i)=1;
        else
            kVector(i)=0;
        end
    end
    %Calc Pk from kVector
    Pk1=sum(kVector)/length(kVector);
    %store results from the above
    PkVector1(q)=Pk1;
end
%average a number of runs to calculate final expected Pk
SSPDr1=mean(PkVector1);
SSPDr1w=SSPDr1*w1e1;

for q=1:10
    clear x2 kVector
    for i=1:20000 %seperate occasions to be simulated
        %x1 and y1 are the OCO errors
        x2=randn*s2e1;
        y2=randn*s2e1;
        r2=sqrt(x2^2+y2^2);
        %If absolute value x1 is inside Wet dimension
        %then a hit has occured
        if r2<=LethalRadius
            kVector(i)=1;
        else
            kVector(i)=0;
        end
    end
    %Calc Pk from kVector
    Pk2=sum(kVector)/length(kVector);
    %store results from the above
    PkVector2(q)=Pk2;
end
%average a number of runs to calculate final expected Pk
SSPDr2=mean(PkVector2);
SSPDr2w=SSPDr2*(1-w1e1);

SSPDrTotal=SSPDr1w+SSPDr2w

%Setup Parameters
mu=0;
s=s1e1;
a=0;
b=LethalRadius;

%Calculate the probability a sample lies within the given range [a:b]
syms r
raypdf=(r/s^2)*exp((-r^2)/(2*s^2));
Prob=int(raypdf,a,b);
SSPD1analytical=double(Prob);
SSPD1analyticalw=SSPD1analytical*w1e1;

```

```

mu=0;
s=s2e1;
a=0;
b=LethalRadius;

%Calculate the probability a sample lies within the given range [a:b]
syms r;
raypdf=(r/s^2)*exp((-r^2)/(2*s^2));
Prob=int(raypdf,a,b);
SSPD2analytical=double(Prob);
SSPD2analyticalw=SSPD2analytical*(1-w1e1);
SSPDrtotalanalytical=SSPD1analyticalw+SSPD2analyticalw

function sse=myfit(params,Input,Actual_Output)
w1f=params(1);
s1f=params(2);
s2f=params(3);
Fitted_Curve=1-w1f.*exp(-Input.*Input/(2*s1f*s1f))-(1-w1f).*exp(-
Input.*Input/(2*s2f*s2f));
Error_Vector=Fitted_Curve - Actual_Output;
% When curvefitting, a typical quantity to
% minimize is the sum of squares error
sse=sum(Error_Vector.^2);
% You could also write sse as
% sse=Error_Vector(:)'*Error_Vector(:);

```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- [1] M. Driels, *Weaponeeing: Conventional Weapon System Effectiveness*. American Institute of Aeronautics and Astronautics, 2004.
- [2] Department of the Army DARCOM Pamphlet, *Engineering Design Handbook Army Weapon Systems Analysis, Part One*, Department of the Army, 1977.
- [3] M. Driels (private communication), 2007.
- [4] M. Driels, "Accuracy Metrics for Precision Weapons," presented at DAWG Meeting, Eglin AFB, FL, 2007.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Fort Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Professor Morris Driels
Naval Postgraduate School
Monterey, California
4. Mr. Tony Rubino
Edwards AFB
Edwards AFB, California